MATLAB COUSE

BY

ENG/MOHAMED NABIL

FACULTY OF ENGINEERING

KFS UNIVERSITY

m.nabil@eng.kfs.edu.eg

2011

Eng.M.Nabil

بسم الله الرحمن الرحيم والصلاة والسلام على نبينا محمد واله وصحبة وسلم:

ما هوMatLab ؟

هو أداة وبيئة تطوير برمجية مخصصة للمهام الحسابية، حيث تتوفر فيه الكثير من الوظائف والدوال الرياضية المبنية داخليا والتي تسهل حل مختلف أنواع المعادلات الرياضية. كما تساعد لغة برمجة على كتابة دوال وبرامج خاصة. بالإضافة للعديد من المميزات الأخرى به.

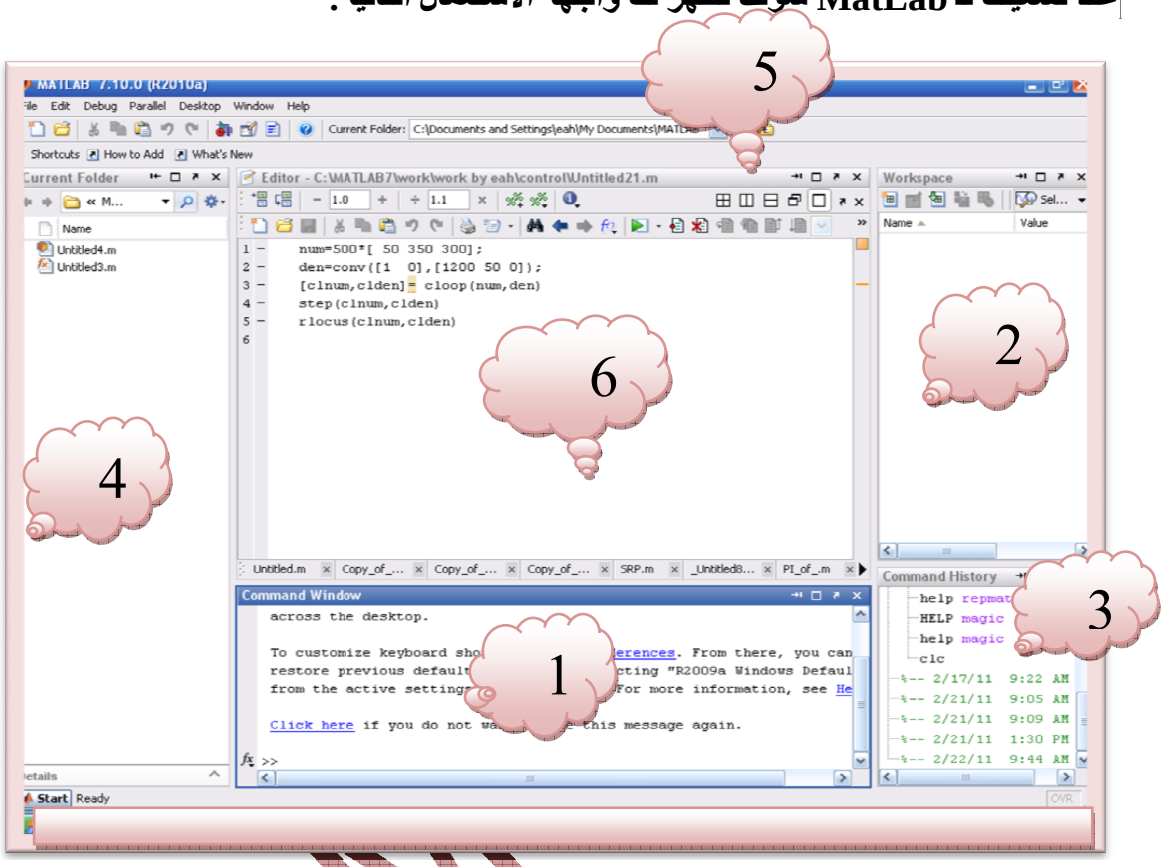
تتضمن استعمالات الـ MatLab المجالات التالية:

- **الرياضيات و الحساب Math and computation**
- **تطوير الخوارزميات Algorithm development**
- **Data acquisition**
- **النمذجة والمحاكاة Modeling, simulation, and prototyping**
- **تحليل واستكشاف وتصوير البيانات Data analysis, exploration, and visualization**
- **الرسوم الهندسية والبيانية Scientific and engineering graphics**
- **بناء واجهات استخدام رسومية للتطبيقات المعدة Application development, including graphical user interface building**

Eng.M.Nabil

واجهة التشغيل:

عند تشغيلك لـ MatLab سوف تظهر لك واجهة الاستعمال التالية:



Eng.M

تتكون الواجهة من مجموعة من الإطارات

1. إطار الأوامر Command Window

ومن خلاله يتم إدخال الأوامر للبرنامج، حيث يظهر المحث على الشكل (>>) ويتم كتابة الأمر بعده، وبما أن لغة Matlab هي لغة مفسرة Interpreted فإننا نحصل على الاستجابة فور الانتهاء من كتابة البرنامج، ولكن يمكن تجنب إظهار النتيجة لكل أمر بإلحاق الأمر بفاصلة منقوطة;

2. إطار منطقة العمل Workspace

حيث يظهر جميع المتغيرات المستعملة في جلسة العمل الحالية.وهو جزء من ذاكرة المتلاب يستخدمه فى تخزين البيانات

3. إطار الأوامر السابقة Command History

حيث يتم عرض جميع الأوامر التي سبق إدخالها في جلسات عمل سابقة.

4. إطار المجلد الحالي Current Directory

في هذا الإطار يتم عرض جميع الملفات الموجودة في مجلد العمل الحالي والذي يكون عادة C:MATLAB 7.10.0work حيث يوجد به البرامج التي سنقوم بتشغيلها.

يمكن تعديل هذا المجلد لأي مجلد آخر من خلال المفتاح (...) المجاور لأسم المجلد في أعلى الإطار، أو من خلال نفس المفتاح الموجود على شريط الأدوات (منطقة رقم 5 في الصورة)

6.إطار المحرر editor

فيها الإطار يمكن كتابة البرامج المختلفة وتنفيذها بطريقة سهلة حيث يتم ترقيم الخطوات بصورة متسلسلة

أما مفتاح **Start** الموجود أسفل الشاشة فهو شبيه لمفتاح **start** في نظام ويندوز، حيث يمكن من خلاله تشغيل بقية الأدوات المرافقة لبيئة.MatLab

ملاحظة :

قد تظهر لديك واجهة الاستعمال مختلفة بعض الشيء عن المعروضة في الصورة، أو قد ترغب أنت في إخفاء بعض الأطر أو جعلها خارج الواجهةundock

لجعل أي أطار خارجيا استعمل مفتاح الموجود على الجانب الأيمن العلوي من الإطار، ولإعادة داخل الواجهة أختار من الإطار(View -> dock (window name

ولإغلاقه استعمل مفتاح

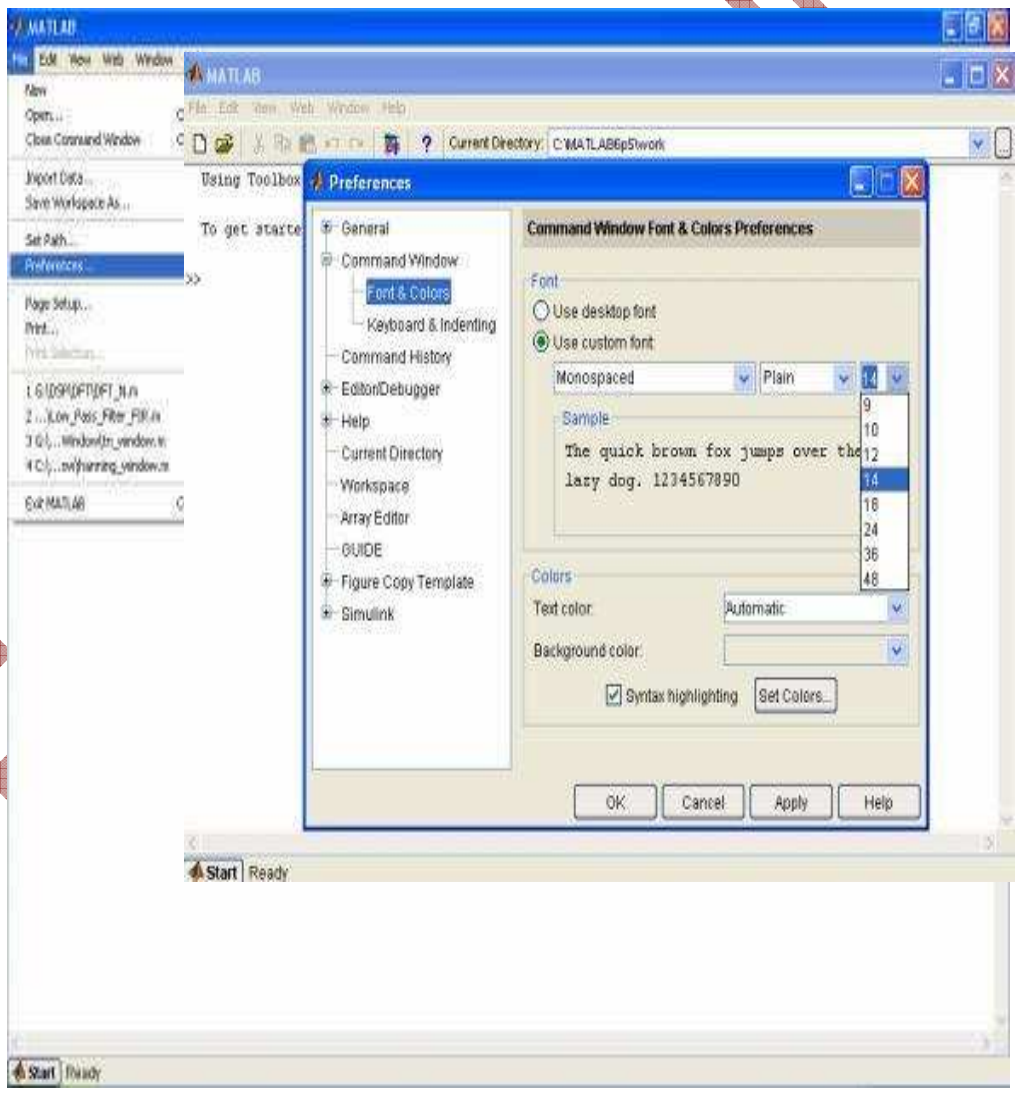
Eng.M.Nabil

توفر الوثائق المرافقة لـ MatLab الكثير من المعلومات المفيدة حول MatLab ويمكنك البدا باستعراضها من خلال اختيار MatLab Help من قائمة Help.

Changing Font Size: لتغير حجم الكتابة:

File> Preferences> Command Window> Font & Colors> use custom font :>

Ex.: Monospaced, Plain, 14



Numbers and Results Representation :تمثيل الارقام وعرض النتائج

```
s = 1 + 2
s =3
fun = sin(pi/4)
fun =
0.7071
format long
fun
fun =
0.70710678118655
format short
fun
fun =
0.7071
realmin
ans =
2.2251e-308
i
ans =
0 + 1.0000i
```

Special variables: **متغيرات ثابته :**

الثابت	قيته
pi	3.14159265...
i	$\sqrt{-1}$
j	مثل العدد التخيلي (i)
eps	2^{-52} رقم صغير جداً
realmin	2^{-1022} أصغر رقم يحسبه البرنامج
realmax	$(2-\varepsilon)^{21023}$ أكبر رقم يحسبه البرنامج
inf	∞ ما لانهاية
NaN	0.0/0.0 or $\infty - \infty =$ غير معرف

Eng.M.Nabil

رموز العمليات: الجمع – الطرح – الضرب –القسمة -الاس

Operation Symbols:

(1)Addition +
(2)Subtraction –
(3)Multiplication *
***(4)Division / or ***
(5)Exponentiation== to the power of

```
X = 47/3
X =
15.6667

Y = 47\3
Y =
0.0638

x = sin(1) - sin(2) + sin(3) - sin(4) + sin(5) -...
sin(6) + sin(7) - sin(8) + sin(9) - sin(10)

 x =
0.7744
```

Complex Numbers: الاعداد المركبة:

Item	Description
Complex(2,-3)	Define a complex number
Abs(x)	Absolute value ;|x|
Angle(x)	Angle of complex number x

Eng.M.Nabil

Conj(x)	Complex conjugate of x
Imag(x)	Imaginary part of a complex number x
Real(x)	Real part of complex number x

```
Command Window
>> z=3+4i;
>> abs(z)

ans =

     5

>> conj(z)

ans =

   3.0000 - 4.0000i

>> angle(z)

ans =

    0.9273

>> imag(z)

ans =

     4

>> real(z)

ans =

     3
```

Trigonometric functions (Radian): الدوال المثلثية دائرى:

Item	Description
Acos(x)	Inverse cosine
Acot(x)	Inverse cotangent
Acsc(x)	Inverse cosecant
Asec(x)	Inverse secant

Asin(x)	Inverse sine
Atan(x)	Inverse tangent
Cos(x)	Cosine
Cot(x)	cotangent
Csc(x)	cosecant
Sin(x)	Sine
Tan(x)	tangent

Trigonometric functions (degree): الدوال المثلثية درجات:

Item	Description
Acosd(x)	**Inverse cosine**
Acotd(x)	**Inverse cotangent**
Acscd(x)	**Inverse cosecant**
Asecd(x)	**Inverse secant**
Asind(x)	**Inverse sine**
Atand(x)	**Inverse tangent**
Cosd(x)	**Cosine**
Cotd(x)	**cotangent**
Cscd(x)	**cosecant**
Sind(x)	**Sine**

Tand(x)	**tangent**

Hyperbolic function: الدوال الزائدية:

Item	Description
Acosh(x)	Inverse cosine
Acoth(x)	Inverse cotangent
Acsch(x)	Inverse cosecant
Asech(x)	Inverse secant
Asinh(x)	Inverse sine
Atanh(x)	Inverse tangent
Cosh(x)	Cosine
Coth(x)	cotangent
Csch(x)	cosecant
Sinh(x)	Sine
Tanh(x)	tangent

Exponential functions: الدوال اللوغارتمية:

item	Description
exp	التابع الاسي
log	اللوغاريتم الطبيعي

log10	اللوغاريتم للاساس 10
log2	اللوغاريتم للاساس2
Sqrt(x)	الجذر التربيعي
nthroot	الجذر من المرتبةn
pow2	2^(x)
expm1	Exp(x)-1
log1p	Log(x+1)

```
Command Window
>> exp(4)

ans =

   54.5982

>> log10(100)

ans =

     2

>> log2(4)

ans =

     2
```

```
>> nthroot(100,2)

ans =

    10

>> sqrt(4)

ans =

     2

>> pow2(4)

ans =

    16
```

Number systems: انظمة العد :

```
Command Window
>> dec2bin(14)

ans =

1110

>> bin2dec('1010')

ans =

    10

>> hex2dec('1a')

ans =

    26

>> dec2hex(26)
```

1-المتجهات في MatLab

يوفر Matlab مجموعة من الأوامر التي تجعل إدخال المتجهات والتعامل معها أكثر سهولة، حيث تشبه الأوامر المستعملة في MatLab أسلوب كتابة المتجهات في الجبر.

خلال هذا الدرس سوف نوضح هذه الأوامر، وكيفية استعمالها.

معلومة:

كلمة MatLab هي اختصار لعبارة matrix laboratory أو مختبر المصفوفات.

إنشاء المتجهات:

ابسط طريقة لتعريف المتجهة هي بكتابة عناصر المتجه يفصل بين كلا منها مسافة ومحصورة بقوسين مربعين [ ]

```
>> a = [1 2 3]

a =

     1     2     3

>>
```

لاحظ أن نتيجة الأمر الذي قمنا بكتابته قد ظهرت لنا مباشرة جرب نفس الأمر السابق ولكن أضف في نهايته فاصلة منقوطة ؛

```
>> a = [1 2 3];
>>
```

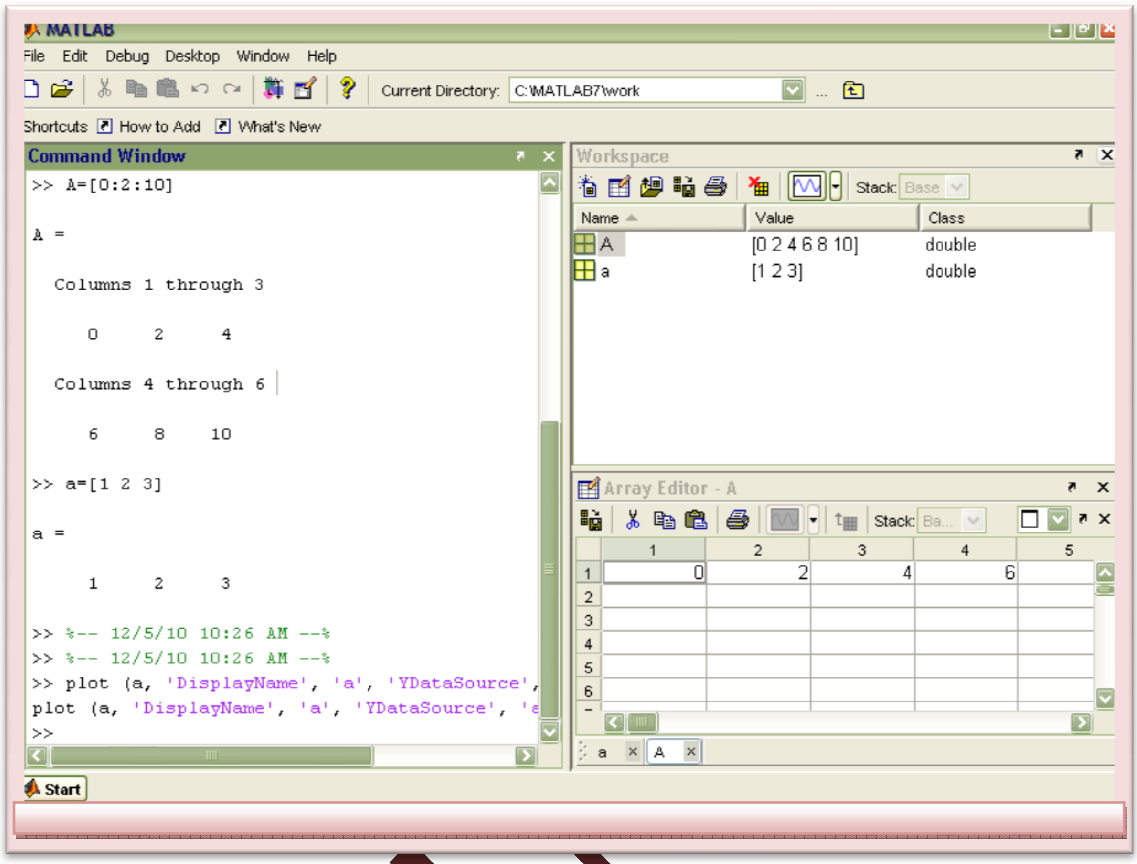
هذه المرة لا تظهر نتيجة الأمر بعده.

كما يمكن تعريف المتجهة من خلال تحديد القيمة الأولى : قيمة الزيادة: القيمة الأخيرة

```
>> A = [0: 2: 10]

A =

     0     2     4     6     8    10
```

في إطار منطقة العمل Workspace لاحظ المتغيرات المعرفة حاليا في جلسة العمل

E.Eng.M.Nabil



كما بالشكل أعلاه يظهر حتى الآن متغيرين هما a و A

MatLab حساس لحالة الأحرف Case-sensitive لذا فإن المتغير a مختلف تماما عن المتغير A.

ملاحظة:

يمكنك كتابة الأمر whos لعرض المتغير المعرفة في جلسة العمل الحالية في إطار الأوامر Command Window

عرض المتجهات:

لعرض محتويات أي متجه نقوم بكتابة اسم المتجه ثم نضغط على مفتاح الإدخال Enter

Eng.M.N

```
>> a

a =

     1     2     3

>>
```

أو يمكن عرض القيمة الثانية في المتجه فقط من خلال الأمر:

```
>> a(2)

ans =

     2

>>
```

لاحظ المتغير الجديد الذي تم أنشاؤه ans. في كل مرة يتم فيها كتابة امر تنتج عنه قيمة بدون تعيين هذه القيمة إلى متغير فأن تلك القيمة سوف تحمل في المتغير ans.

لعرض أول 4 قيم بالمتجه، أو لعرض القيمة الأولى والرابعة فقط:

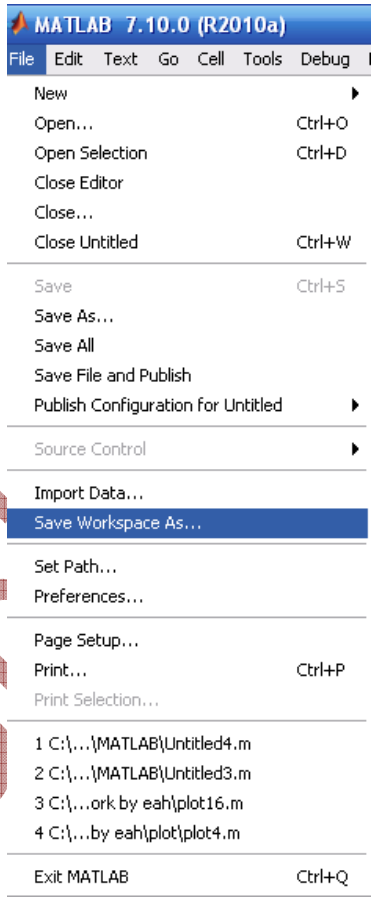
```
>> A(1:4)

ans =

     0     2     4     6

>> A(1:3:4)

ans =

     0     6
```

الآن وبعد أن تعرفنا على كيفية التعامل مع المتجهات في بيئة MatLab ، نختتم الدرس بحفظ جلسة العمل الحالية للرجوع لها في أي وقت آخر

<u>حفظ واسترجاع جلسة العمل:</u>

لحفظ جلسة العمل WorkSpace أختار من قائمة File ->

Save workspace As



ومن خلال مربع حوار Save As أختار اسم مناسب لملف جلسة العمل مثلا: MyFirstMat

الملف سوف يحفظ في مجلد العمل والذي يكون عادة مجلد C:MATLAB7work ويعطى الامتداد MAT

وعند تشغيل MatLab مرة ثانية يمكن استعادة ملف جلسة العمل من خلال File -> Open ثم اختار الملف ذو الامتداد Mat الذي حفظت به جلسة العمل.

Eng.M.Nabil

المصفوفات فى MatLab:

خلال هذا الدرس سوف نستكمل دراسة المزيد من أوامر Matlab والمتعلقة بإنشاء المصفوفات والتعامل معها.

إنشاء المصفوفات:

طريقة تعريف المصفوفات في MatLab قريبا جداً إلى طريقة تعريف المتجهات، نبدأ مباشرة مع أول مثال:

```
>> D = [1 2 3; 4 5 6; 7 8 9]

D =

1 2 3
4 5 6
7 8 9
```

لاحظ الفرق بين فصل الأعداد بمسافة أو فاصلة منقوطة، جرب هذا الأسلوب كذلك:

```
>> D = [ 1 2 3;
4 5 6;
7 8 9]

D =
1 2 3
4 5 6
7 8 9
>>
```

كما يوجد عدد من الدوال لإنشاء مصفوفات خاصة:

1. دالة pascal لإنشاء مصفوفة متناظرة symmetric
2. دالة magic لإنشاء مصفوفات يتساوى فيها مجموع كل الصفوف والاعمدة.
3. دالة zeros لإنشاء مصفوفة صفرية.
4. دالة ones لإنشاء مصفوفة كل عناصرها تساوي 1

Eng.M.Nabil

لاحظ الامثلة التالية:

```
>> P = pascal(3)

P =

1 1 1
1 2 3
1 3 6

>> M= magic(3)

M =

8 1 6
3 5 7
4 9 2

>> z= zeros(2, 3)

z =

0 0 0
0 0 0

>> o = ones(2, 4)

o =
1 1 1 1
1 1 1 1
>>
```

Eng.M.Nabil

العمليات الحسابية على المصفوفات:

كما ذكرنا سابقا فإن MatLab يجعل التعامل مع المتجهات والمصفوفات أكثر سهولة، جرب الأمثلة التالية:

```
>> Sum = D + P

>> Sub = P - D

>> D = D + 2
```

```
>> P2 = P * 2

>> Mult1 = P * D

>> Mult2 = P .* D

>>  Div=  D  /  P
```

الأمر الأول: يجمع كلا من P و D وينتج عنه المصفوفةSum

الأمر الثاني: ناتج طرح D من P في المصفوفةSub

الأمر الثالث: يضيف 2 إلى كل عنصر من عناصر المصفوفةD

الأمر الرابع: ينتج عنه مصفوفة Mult1 والتي يحفظ بها ناتج ضرب P في D

الأمر الخامس: *لاحظ النقطة قبل علامة الضرب* (هذا الأمر سينتج عنه مصفوفة Mult2 والتي هي عبارة عن حاصل ضرب كل عنصر في P في العنصر المقابل له فيD)

الأمر السادس:ينتج عنة المصفوفة Div والتي يحفظ بها ناتج قسمة D/P

جرب أيضا الأمرين التالين ولاحظ الفرق في الناتج

```
>> M

M =

8 1 6
3 5 7
4 9 2

>> MM = M ^ 2

MM =

91 67 67
67 91 67
67 67 91
```

Eng.M.Nabil

```
>> M2 = M .^ 2

M2 =

64 1 36
9 25 49
16 81 4

>>
```

M ^ 2 يعني ضرب المصفوفة في نفسها

M .^ 2 يعني ضرب كل عنصر في المصفوفة في نفسه.

لإيجاد محورة المصفوفة

Transpose

```
>> M'

ans =

8 3 4
1 5 9
6 7 2
```

لإيجاد معكوس المصفوفةInverse

```
>> inv(M)

ans =

0.1472 -0.1444 0.0639
-0.0611 0.0222 0.1056
-0.0194 0.1889 -0.1028

>>
```

للتعرف على حجم المصفوفة

```
>> size(z)

ans =
```

Eng.M.Nabil

```
2 3
 >> size(o)

ans =

2 4

>>
```

العدد الأول يمثل عدد الأسطر والثاني عدد العمود

Eng.M.Nabil

كثير الحدود فيMatLab

أهداف الدرس:

التعرف على كيفية تمثيل كثير الحدود في MatLab ، وكيفية التعامل معها.

يوفر Matlab عدد من الدوال المبنية داخليا لتسهيل التعامل مع كثير الحدود Polynomials ، حيث يتم تمثيلها كمتجه، مثلا لتمثيل معادلة كثير الحدود التالية:

$$S^4 + 3S^3 - 15S^2 - 2S + 9$$

نعرف المتجه التالي:

```
>> x = [1 3 -15 -2 9]

x =

1 3 -15 -2 9
```

$S^4 - 2$ **كذلك لتمثيل**

```
>> Z = [1 0 0 0 -2]

Z =

1 0 0 0 -2
```

حساب قيمة كثير الحدود عند قيمة محددة:

لكي نحسب قيمة كثير الحدود الأول x عند قيمة s=3 مثلا، يمكن استعمال دالةpolyval

Eng.M.Nabil

```
x =

     1     3   -15    -2     9

>> polyval(x, 3)

ans =

    30

>>
```

احسبها وتأكد من الناتج:

إيجاد جذور كثير الحدود:

يقصد بالجذور قيم المتغير s التي تجعل القيمة الكلية للمعادلة تساوي 0

```
>> roots(x)

ans =

   -5.5745
    2.5836
   -0.7951
    0.7860

>>
```

والعكس:

يعني لاكتشاف معادلة كثير الحدود لجذور معلومة، الدالة هنا هي poly

```
>> poly(ans)

ans =

    1.0000    3.0000  -15.0000   -2.0000    9.0000

>>
```

ضرب وقسمة كثير الحدود:

لضرب معادلتين كثير حدود في بعضهما استعمال دالة conv وللقسمة الدالةdeconv

```
>> x

x =

     1     3   -15    -2     9

>> z

z =

     1     0     0     0    -2

>> mu = conv(x, z)

mu =

     1     3   -15    -2     7    -6    30     4   -18

>> [d, r] = deconv(mu, x)

d =

     1     0     0     0    -2


r =

     0     0     0     0     0     0     0     0     0
```

عند استعمال deconv لقسمة كثيري حدود فأنه ينتج متجهين:

- **الأول d ناتج القسمة.**
- **الثاني r باقي القسمة) وفي المثال السابق كان الباقي من القسمة متجه صفري.(يمثل عدد الأعمدة**

Eng.

أوامر مفيدة فيMatLab

وقفة قصيرة من الأمور الرياضية ودوالها التي تكلمنها عنها في الدروس السابقة، لنتعلم المزيد عن كيفية استعمال matlab والأوامر الأساسية به .

مسح إطار الأوامر:

أثناء عملنا قد نرغب من وقت لآخر في مسح كل ما هو موجود على إطار الأوامر. يوجد طريقتين لذلك:

1. إذا كنت من محبي استعمال الفأرة أختار Edit-> Clear Command Window
2. أما إذا كنت تفضل استعمال لوحة المفاتيح فأكتب clc ثم Enter.

ملاحظة:

مسح إطار الأوامر لن يحذف المتغيرات التي تم تعريفها خلال جلسة العمل، أنظر لإطار جلسة العمل Workspace ستجد أن المتغيرات لم تتغير أو تحذف. يمكنك أيضا استعمال الأمر hos لعرض المتغير المعرفة في جلسة العمل الحالية للتأكد، إذا كان إطار جلسة العمل غير ظاهر لديك.

حذف جميع المتغيرات المعرفة في جلسة العمل:

وهنا أيضا لدينا طريقتين:

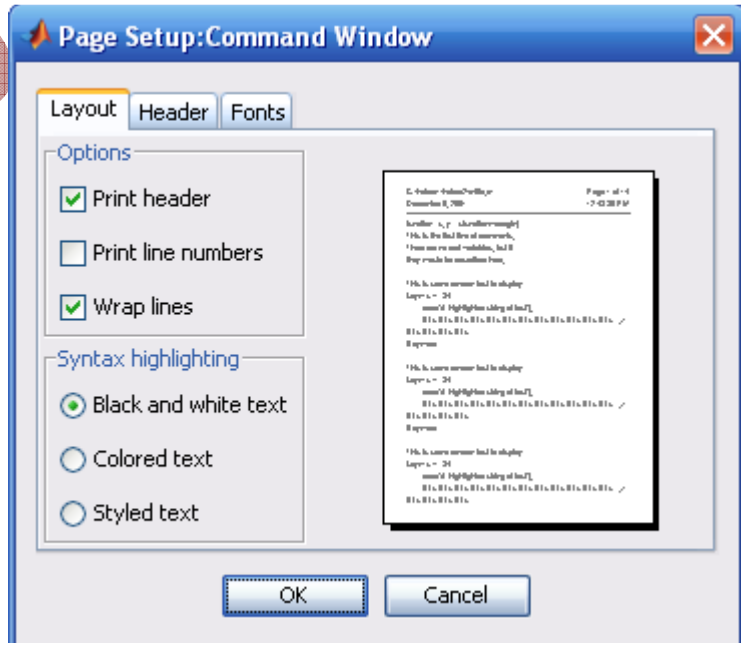
1. إذا كنت من محبي استعمال الفأرة أختار Edit-> Clear Workspace
2. أما إذا كنت تفضل استعمال لوحة المفاتيح فأكتب clear ثم Enter.

طباعة محتويات إطار الأوامر:

لطباعة محتويات إطار الأوامر كاملة اختار File -> Print

ولطباعة الجزء المحدد فقط من الإطار أختار File -> Print selection

وللتحكم في تنسيق المخرجات من الطباعة اختار File -> Page Setup حيث تظهر لك مربع حوار page setup الذي يمكن من خلاله التحكم في تنسيق الصفحة مثل ظهور رأس الصفحة Headerأو لا ومحتويات هذا الرأس، ظهور أرقام للأسطر، والخطوط fonts المستعملة إثناء الطباعة.



Eng.M.Nabil

ملاحظات مفيدة :

- **خلال عملك على matlab تذكر أنه حساس لحالة الأحرفcase sensitive ، لذلك فإن Clear ليست مثل clear على سبيل المثال .**
- **يمكن أن تكتب أكثر من أمر على سطر واحد في MatLab شرط أن تفصل بينهما بفاصلة منقوطة.**

```
>> A = [1 2 3 4 5]; B = [6 7 8 9 10];
>> C= A + B

C =

7 9 11 13 15

>>
```

كما يمكن كتابة الأمر الواحد على سطرين منفصلين، (خاصة إذا كان عرض الشاشة لا يتسع له) بأن نضع ثلاث نقاط (...) عند نهاية السطر الأول.

```
>> D = [ 2 5 2 4 1 66 8 44 88 66 ...
5 7 44 88 44 787 56 66 4]

D =

Columns 1 through 12

2 5 2 4 1 66 8 44 88 66 5 7

Columns 13 through 19

44 88 44 787 56 66 4

>>
```

خلال العمل على MatLab فإن الأوامر التي تكتبها في أطار الأوامر تحفظ في حافظة الـ Historyوقد تسأل ما الفائدة من هذا؟

الفائدة منه أنه يمكنك إعادة استدعاء إي من هذه الأوامر السابقة وتنفيذها من جديد، وذلك من خلال الضغط المتكرر على مفتاح السهم للأعلى حتى تصل إلى الأمر الذي تريد تكراره، وذلك دون الحاجة إلى إعادة كتابته مرة ثانية.

Eng.M.Nabil

- للحصول على المساعدة حول أي أمر أو دالة في MatLab مباشرة في إطار الأوامر أكتب help ثم اسم الأمر أو الدالة وسوف تظهر لك كل المعلومات التي تريدها حول ذلك الامر، جرب مثلاhelp sin

4-البرمجة فيMatLab ("Loops in MatLab")

كما ذكرنا في الدرس الأول من هذه السلسلة فإن MatLab هو بيئة تطوير برمجية تحوى العديد من الدوال الجاهزة، بالإضافة إلى إمكانية كتابة برامج ودوال خاصة بنا حسب الحاجة. خلال هذا الدرس سوف نتعرف على الأوامر البرمجة في.MatLab

الجملة الشرطية: if

تستخدم للاختيار بين أمرين حسب شرط محدد

الصيغة العامة:

```
if <condition

  <program1>

else

  <program2>

end
```

في حالة تحقق الشرط condition يتم تنفيذ الكود في program1 وإذا لم يتحقق الشرط يتم تنفيذ الكود فيprogram2

مثال:

```
>> if n < 0
  disp('n is negative')
else
  disp('n is positive')
end
n is positive
>> n

n =

71

>>
```

Eng.M.Nabil

يمكن أن تأخذ جملة if شكلا أكثر تداخلا باستعمال أكثر من مستوي لـelseif

```
if expression1
  statements1
elseif expression2
  statements2
else
  statements3
end
```

أو يمكن استعمال جملة switch التي لها نفس العمل

جملة switch

الصيغة العامة:

```
switch switch_expr
  case case_expr
     statement,...,statement
  case {case_expr1,case_expr2,case_expr3,...}
     statement,...,statement
  ...
  otherwise
     statement,...,statement
```

حيث:

switch_exprهو المتغير (أو التعبير) الذي سيتم اختبار قيمته.

case_exprأحد القيم التي يمكن أن يأخذها المتغيريمكن أن تتضمن الحالة الواحدة أكثر من قيمة، وإذا كانت القيمة للـ switch_expr غير مدرجة في اي حالة ينتقل التنفيذ للقسم otherwise

الحلقات التكرارية:

عندما نرغب في تكرار أمر معين (أو أكثر) عدة مرات، فإن أفضل طريقة لعمل ذلك هو بوضع هذا الأمر داخل حلقة تكرارية.

Eng.M.Nabil

في MatLab يوجد نوعين فقط من الحلقات التكرارية:

1. حلقة for

وتستخدم عندما يكون المطلوب هو التكرار لعدد محدد من المرات.

الصيغة العامة

```
for variable = expression
  statement
  ...
  statement
end
```

مثال: حلقة بسيطة سوف تتكرر 4 مرات

```
>> for j=1:4

j =

1

j =

2

j =

3

j =
4
end
```

Eng.M.Nabil

2. حلقة while

حيث يكون التكرار هنا مرتبط بتحقق شرط ما، فإذا لم يعد الشرط محقق تنتهي الحلقة

الصيغة العامة:

```
while expression
  statements
end
```

مثال :هذا البرنامج يوجد أول عدد صحيح مضروبه !n مكون من 100 خانة عشرية

```
>> n = 1;

while prod(1:n) < 1e^100
n = n + 1;
end
>> n

n =

70
```

Eng.M.Nabil

ملاحظة:

لغة MatLab هي لغة مفسرة Interpreted أي أن كل أمر يتم ترجمته للحاسوب قبل تنفيذه مباشرة، لذا فإن استعمال الحلقات التكرارية سوف يجعل البرنامج أكثر بطأ، ويفضل استعمال الاوامر والدوال الجاهزة لـ MatLab كلما أمكن ذلك.

break :

يستخدم هذا الأمر لإيقاف تنفيذ حلقة تكرارية وإعادة التحكم للبرنامج أو للحلقة الخارجية عند وجود حلقات متداخلة.

continue:

يقوم هذا الأمر بوقف التكرار الحالي للحلقة iteration ويبدأ في التكرار التالي له.

أمثلة:

برنامج لايجاد جذور معادلة تربيعية

Disp.→this program is used to solve the quadratic eq^{n}

Disp→ $Ax^2+Bx+c=0$

a=input→enter the value of A:___

b=input→enter the value of B:___

c=input→ enter the value of C:___

$D=B^2- 4AC$

0 **Non zero**

X1=x2=-b/2a

X_1= (-B+sqrt (d))/ (2*a))

X_2= (-B-sqrt (d))/ (2*a)

Eng.M.Nabil

```
clc;clear all;
disp('this program is used to solve the quadratic eqn');
disp('Ax2+Bx+c=0');
a=input('enter the value of A:___');
b=input('enter the value of b:___');
c=input('enter the value of c:___');
d=b^2-4*a*c;
if (d==0)
x= (-b/2*a);
disp ('equal roots');
disp (['x1=x2=',num2str(x)])
else
x1= ((-b+sqrt (d))/ (2*a));
x2= ((-b-sqrt (d)) / (2*a));
disp ('distinct roots');
disp (['x1=       ', num2str(x1)]);
disp (['x2=       ', num2str(x2)]);
end
```

If R = 10 Ohms and the current is increased from 0 to 10 A with increments of 2A, write a MATLAB program to generate a table of current, voltage and power dissipation.

Solution

```
R=10;      % Resistance value
i=(0:2:10); % Generate current values
v=i.*R;     % array multiplication to obtain voltage
p=(i.^2)*R;  % power calculation
sol=[i v p]  % current, voltage and power values are printed
```

```
sol =
    Columns 1 through 6
    0     2     4     6     8    10
    Columns 7 through 12
    0    20    40    60    80   100
    Columns 13 through 18
    0    40   160   360   640  1000
```

Columns 1 through 6 constitute the current values, columns 7 through 12 are the voltages, and columns 13 through 18 are the power dissipation values.

Eng.M.Nabil

Plotting

الرسم باستخدام متلاب

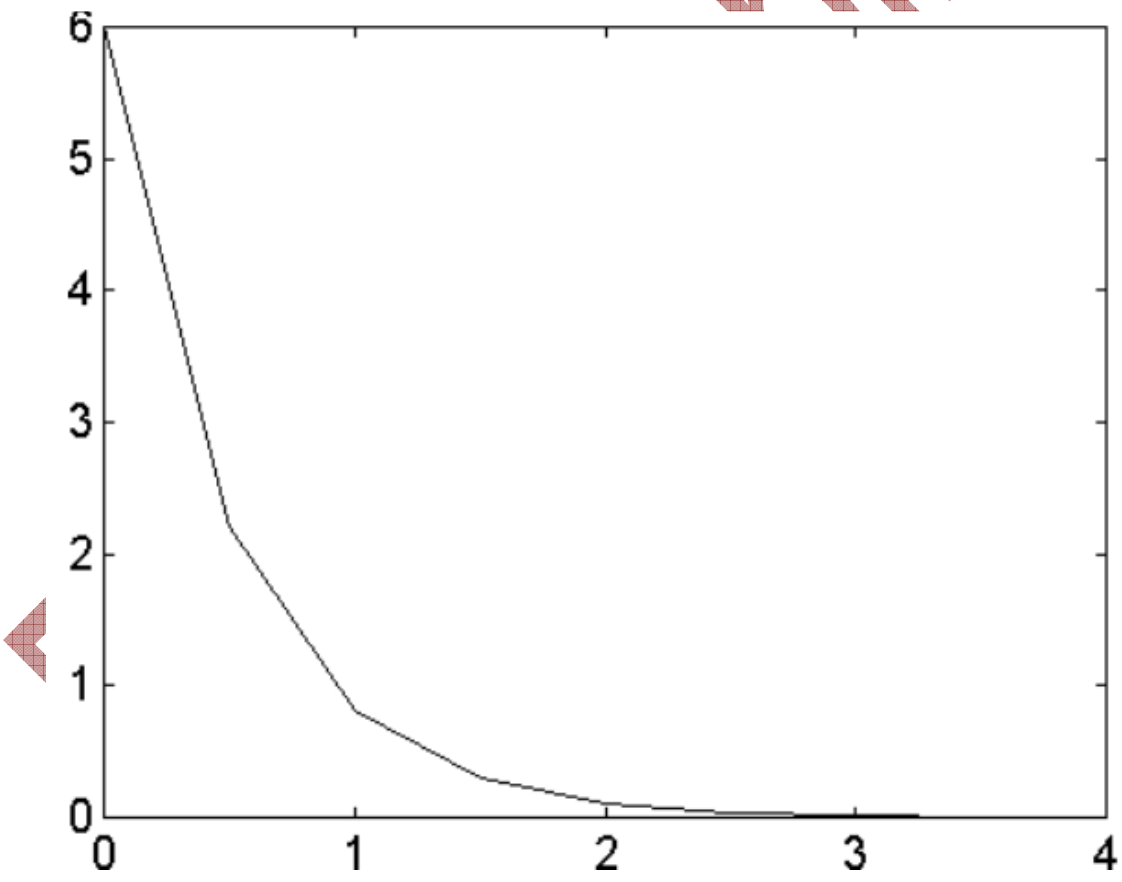
الامر

plot(x,y)

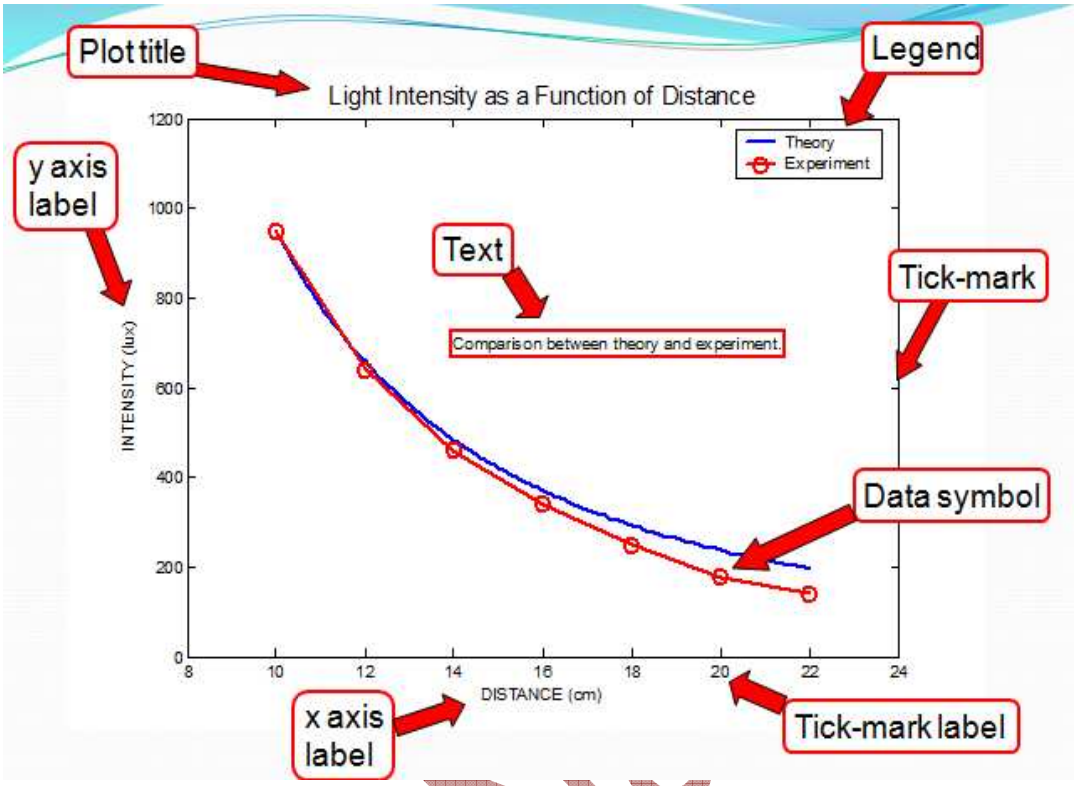
بحث كلا من x&y متغيرات او متجهات متساويه العدد

مثال:

```
t = 0:0.5:4;
y = 6*exp(-2*t);
plot(t,y)
```



ملاحظات هامة:




```
plot(x,y,'line specifiers')
```

Line Style	Specifier	Line Color	Specifier	Marker Type	Specifier
Solid	-	red	r	plus sign	+
dotted	:	green	g	circle	o
dashed	--	blue	b	asterisk	*
dash-dot	-.	Cyan	c	point	.
		magenta	m	square	s
		yellow	y	diamond	d
		black	k		

امثلة:

plot(x,y)	A solid blue line connects the points with no markers.
plot(x,y,'r')	A solid red line connects the points with no markers.
plot(x,y,'--y')	A yellow dashed line connects the points.
plot(x,y,'*')	The points are marked with * (no line between the points.)
plot(x,y,'g:d')	A green dotted line connects the points which are marked with diamond markers.

لرسم اكتر من منحنى على نفس الرسمه نستخدم الامر التالى:

Plot(x1,y1,x2,y2,x3,y3)

حيث ان كلا من x_1&y_1 و x_2&y_2 و x_3&y_3

لعمل عنوان للرسمة:

title(' Text ')

نكتب مكان كلمة textالعنوان الذى تريده

لتسمية كلا من محور x ومحو y:

xlabel(' Text ')

ylabel(' Text ')

legend('Text1','Text2','Text3')

Eng.M.Nabil

مقدمة فى ال SIMULINK

ثانيا :لنتعرف ما هو ال Simulink؟؟

ال Simulinkهو برنامج للنمذجة و المحاكاة و تحليل الانظمة الديناميكية سواء كانت خطية او غير خطية و يقوم أيضا بنمذجة الانظمة سواء فى الزمن المستمر او فى الزمن الغير مستمر .

وباستخدام ال simulinkيمكنك بناء نماذج من البداية او التعديل على انظمة موجودة بالفعل والفائدة من ذلك هو دراسة خصائص نظام التحكم اوالمنظومة قبل البدء فى التنفيذ حتى نحدد مدى استجابة النظام لما نقوم بعمله وهو الحاكم وهل نظام التحكم الموجود سيعطى احسن استجابة وأقل اخطاء ام لا ؟

والSimulink ليس قاصرا على التحكم وتطبيقاته وانما يحتوى على مجموعة من البلوكات والتى تغطى أغلب تطبيقات الهندسة الميكانيكية والكهربية وهندسة الطيران .

ويعتبر ال SIMULINKاداة ممتازة لى Model-Based Designوهذا معناه ان البرنامج ليس فقط قاصرا على الانظمة المثالية ولكن يمكنك ايضا من نمذجة انظمة حقيقة والتى يوجد بها عوامل موثرة لجعلها غير خطية nonlinearمثل الاحتكاك ومقامة الهواء وانزلاق التروس والظواهر الطبيعية الاخرى .

كما يوجد فى البرنامج العديد من النماذج Demoلاغلب التطبيقات يمكنك استخدامها او التعديل عليها .

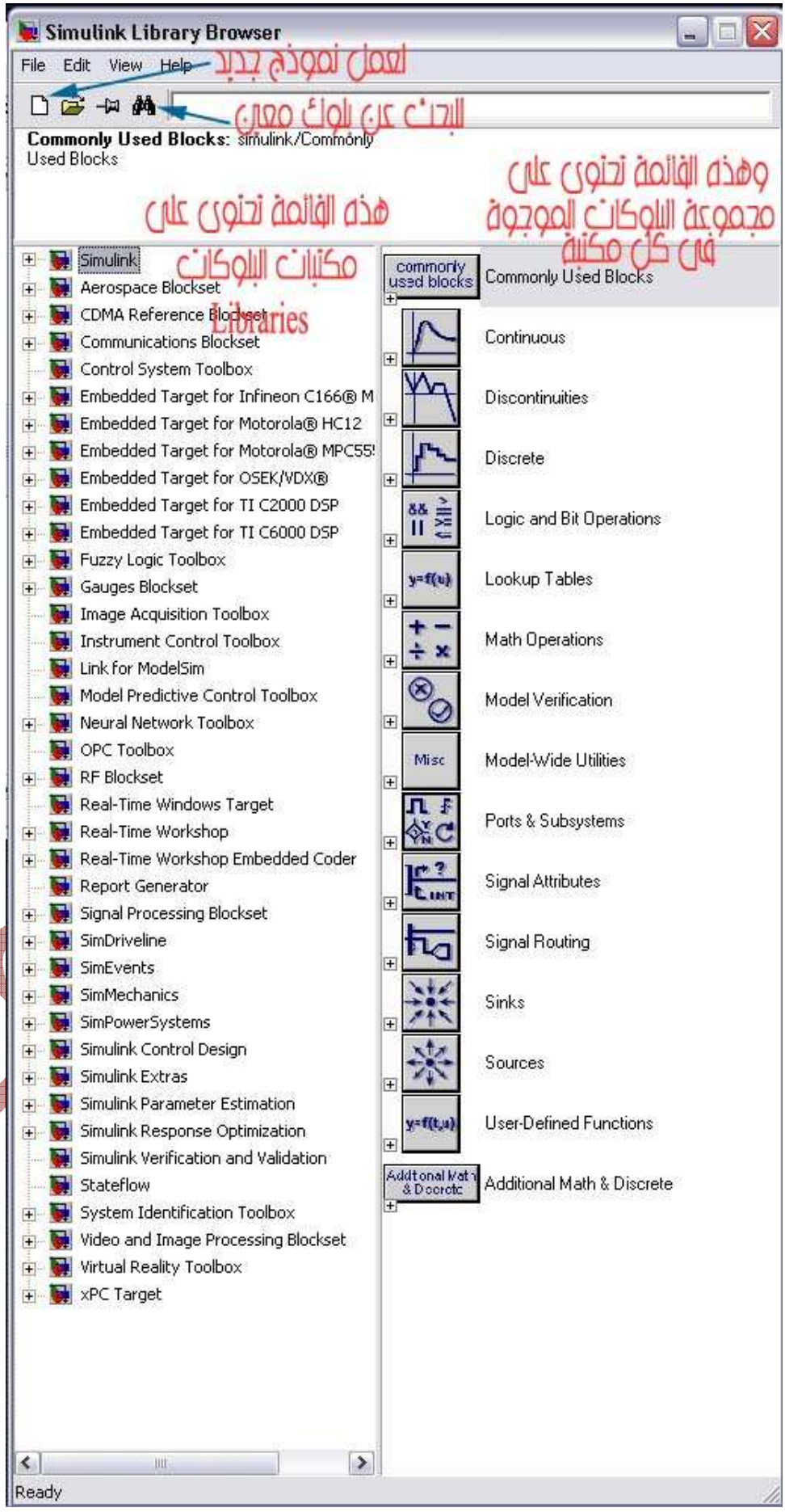
والتعامل مع ال simulinkسهل جدا فهو يوفر بما يسمى graphical user interface (GUI)فى بناء النماذج حيث تقوم بسحب البلوكات التى تريدها الى صفحة النموذج وتقوم بتوصيلها بطريقة سهلة و يمكنك ايضا تغير خصائص البلوكات الموجودة بالضغط عليها بالماوس وتعديل خصائصها كما يمكنك ايضا عمل البلوكات الخاصة بك و يكون هذا باستخدام ما يسمى بى S-function وسوف نتعرض له لاحقا .

وبعد بناء النموذج نقوم بتشغيل النموذج لعمل ال simulationويمكنك اختيار خصائص ال simulationوطريقة التكامل وهذا يكون فى non-real timeكما يمكننا ايضا عمل محاكاة للنماذج فى ال real timeوهذا باستخدام مجموعة البلوكات الموجودة فى البرنامج باسم Xpc Targetو Real Time Workshop وسوف نتعرض لهذا لاحقا .

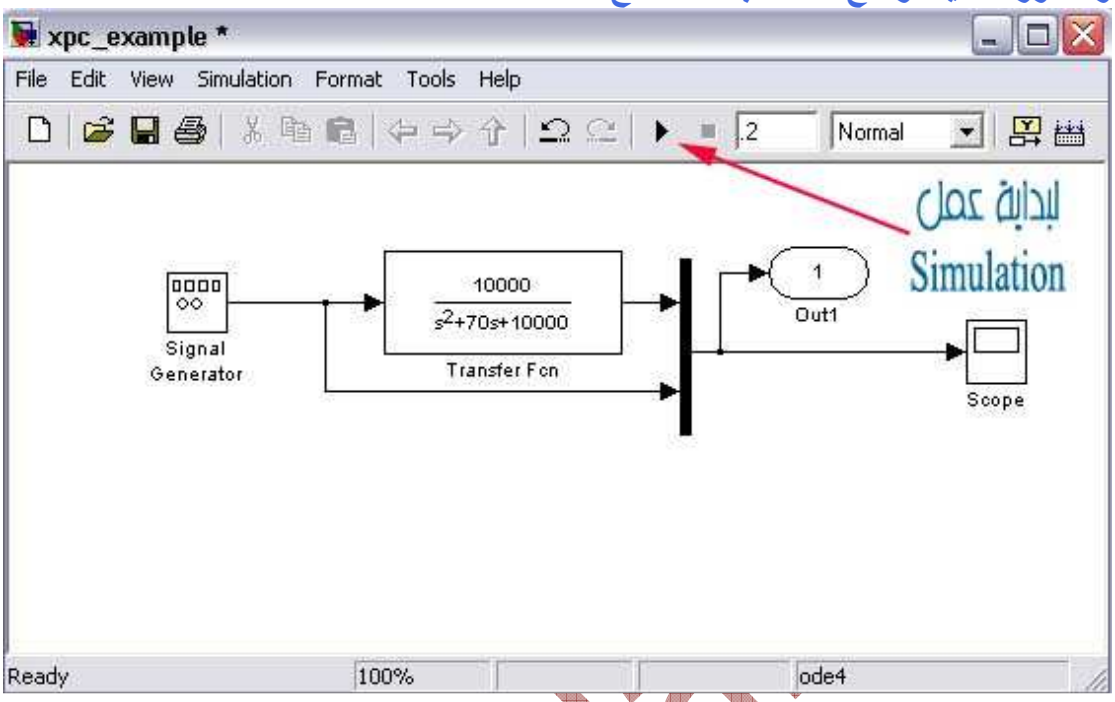
ويمكننا التحكم فى ال Simulationمن خلال سطر أوامر الماتلاب وهذا يكون مفيد جدا فى حالة الرغبة لعمل Simulationلأكثر من نموذج و ويمكن تخزين النتائج و استخدامها مع ال Toolboxsالموجودة فى الماتلاب .

والصورة الاتية توضح الواجهة الرئيسية للبرنامج

Eng.M.Nabil


Simulink Library Browser
File Edit View Help
لعمل نموذج جديد
البحث عن بلوك معين
Commonly Used Blocks: simulink/Commonly Used Blocks
هذه القائمة تحتوى على مكتبات البلوكات Libraries
وهذه القائمة تحتوى على مجموعة البلوكات الموجودة فى كل مكتبة
Simulink
Aerospace Blockset
CDMA Reference Blockset
Communications Blockset
Control System Toolbox
Embedded Target for Infineon C166® M
Embedded Target for Motorola® HC12
Embedded Target for Motorola® MPC55
Embedded Target for OSEK/VDX®
Embedded Target for TI C2000 DSP
Embedded Target for TI C6000 DSP
Fuzzy Logic Toolbox
Gauges Blockset
Image Acquisition Toolbox
Instrument Control Toolbox
Link for ModelSim
Model Predictive Control Toolbox
Neural Network Toolbox
OPC Toolbox
RF Blockset
Real-Time Windows Target
Real-Time Workshop
Real-Time Workshop Embedded Coder
Report Generator
Signal Processing Blockset
SimDriveline
SimEvents
SimMechanics
SimPowerSystems
Simulink Control Design
Simulink Extras
Simulink Parameter Estimation
Simulink Response Optimization
Simulink Verification and Validation
Stateflow
System Identification Toolbox
Video and Image Processing Blockset
Virtual Reality Toolbox
xPC Target
commonly used blocks
Commonly Used Blocks
Continuous
Discontinuities
Discrete
Logic and Bit Operations
Lookup Tables
Math Operations
Model Verification
Misc
Model-Wide Utilities
Ports & Subsystems
Signal Attributes
Signal Routing
Sinks
Sources
User-Defined Functions
Additional Math & Discrete
Ready

والصورة الاتية توضح صفحة بناء النماذج



ثانيا :سنرى كيفية بناء النماذج Building a Model
سنقوم الان بعمل نموذج بسيط لنظام معين و يتكون هذا النموذج من
Signal generator

حيث يقوم بتوليد العديد من أنواع أل signalsلتطبيقها على النظام الموجود وسنستخدم منها .square wave

Transfer function

وهى تمثل النظام الموجود لدينا وهو نظام بسيط من الدرجة الثانية

Scope

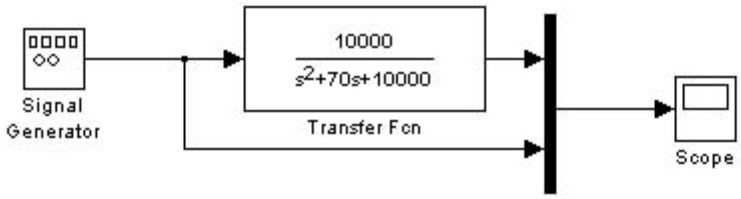
ويستخدم فى عرض نتائج عملية ال Simulation

Mux block

ويستخدم هنا لعرض اكثر من signalفى نفس ال scope

Eng.M.Nabil

وفى الصورة التالية نرى الشكل العام للنموذج :



وللبدء فى العمل

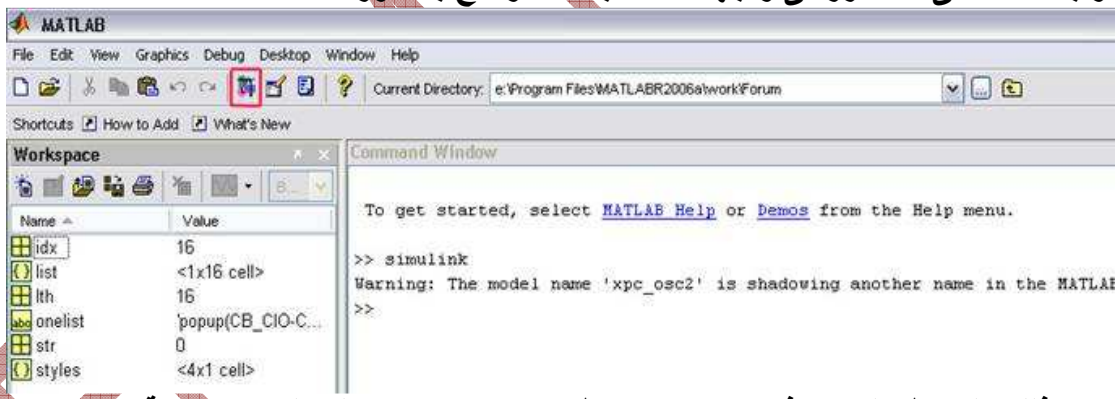
اولا سنقوم بفتح نموذج فارغ ::

لفتح برنامج ال simulinkقم بالكتابة فى سطر أوامر الماتلاب ما يلى

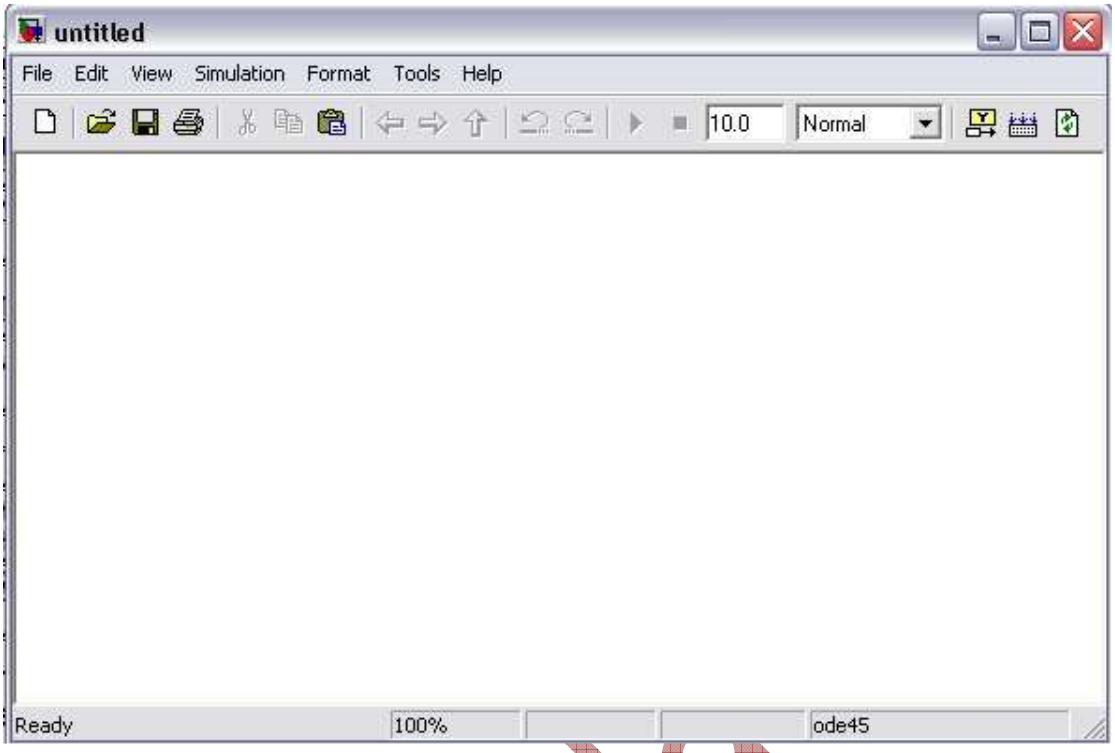
رمز:

```
>>simulink
```

او بالضغط على هذا الزر فى واجهة الماتلاب كما موضح بالصورة



وبعد ذلك بالضغط على نموذج جديد كما لاحظنا سابقا وسيكون على الصورة الاتية :



والان سنقوم بمرحلة اضافة البلوكات المطلوبة

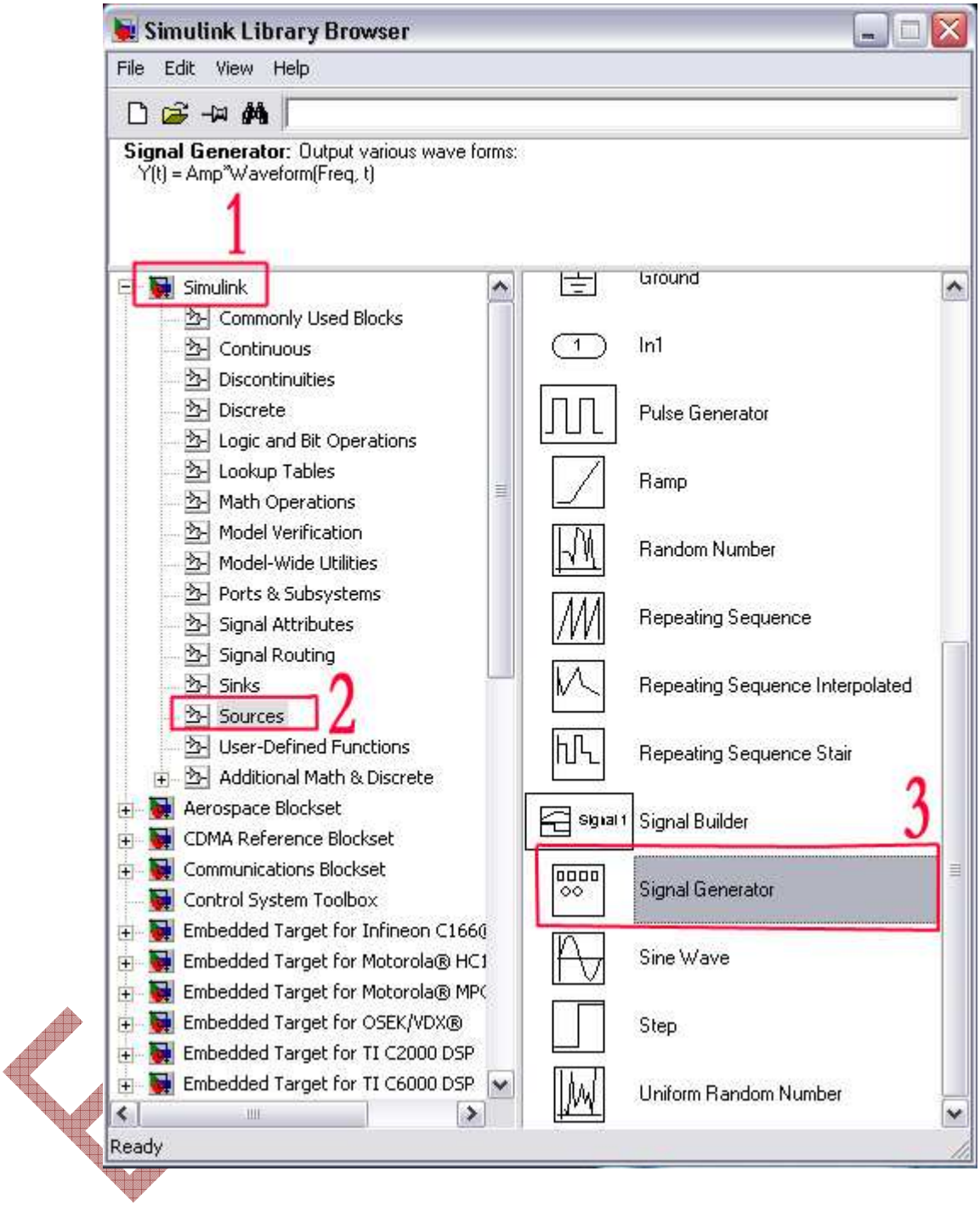
اولا أضافة ال Signal generator

قم بالضغط على قائمة Simulinkلتفتح لك مجموعة من الاقسام

اختار منها Sources
وستفتح لك على اليمين مجموعة البلوكات الموجودة فى هذا القسم أختار منها signal generator
وقم بالضغط على الماوس وسحب البلوك الى النموذج

والخطوات السابقة موضحة فى الصورة التالية

Eng.M.Na



ثانيا

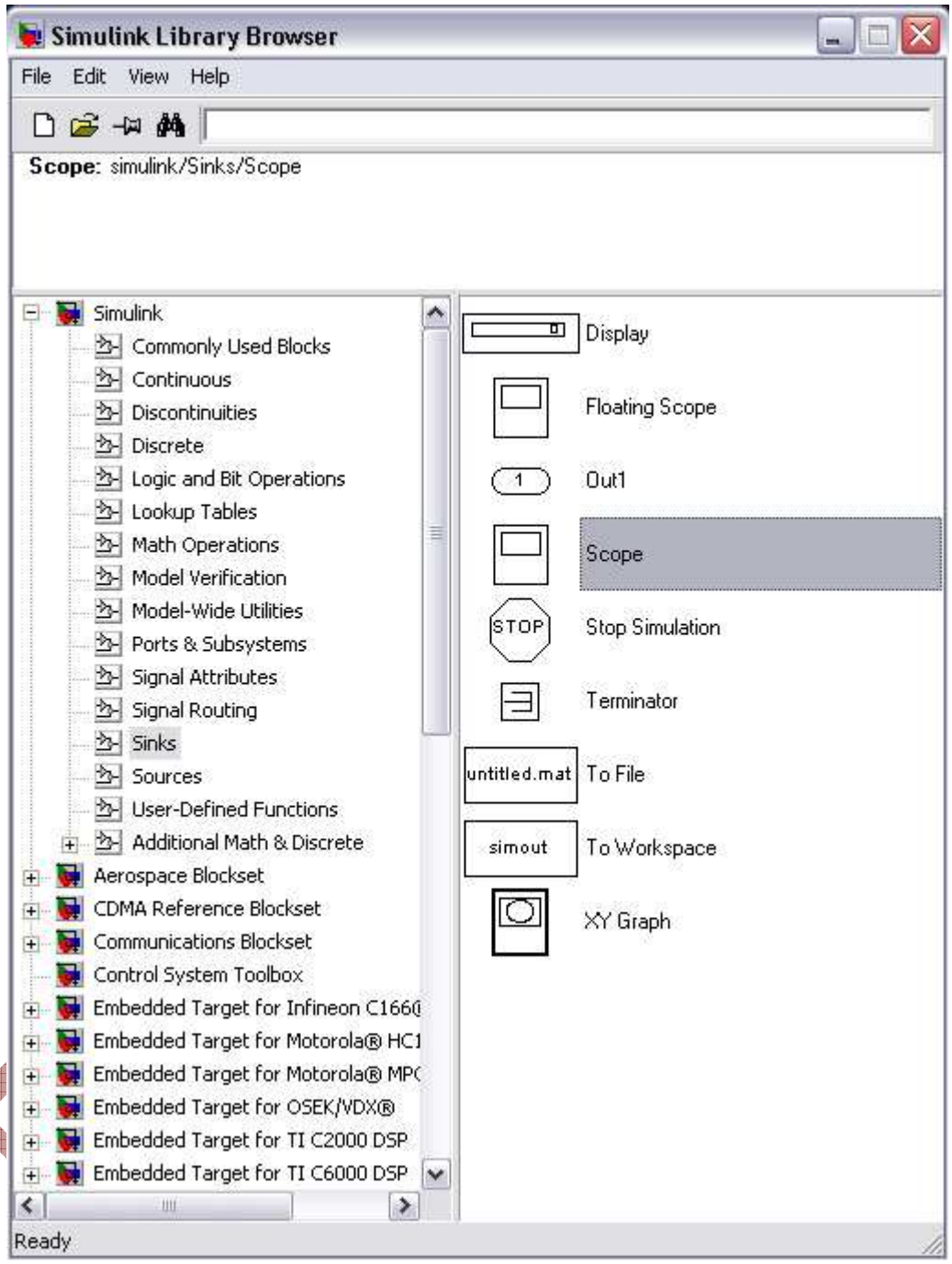
اضافة ال scope:

1) **قم بالضغط على قائمة simulinkلتفتح لك مجموعة من الاقسام**

2) **أختار منها sinks**

3) **أختار من اليمين scope**
4) **قم بالضغط على الماوس وسحب البلوك الى النموذج**

والخطوات السابقة موضحة فى الشكل الاتى



ثالثا :لاضافة ال transfer function

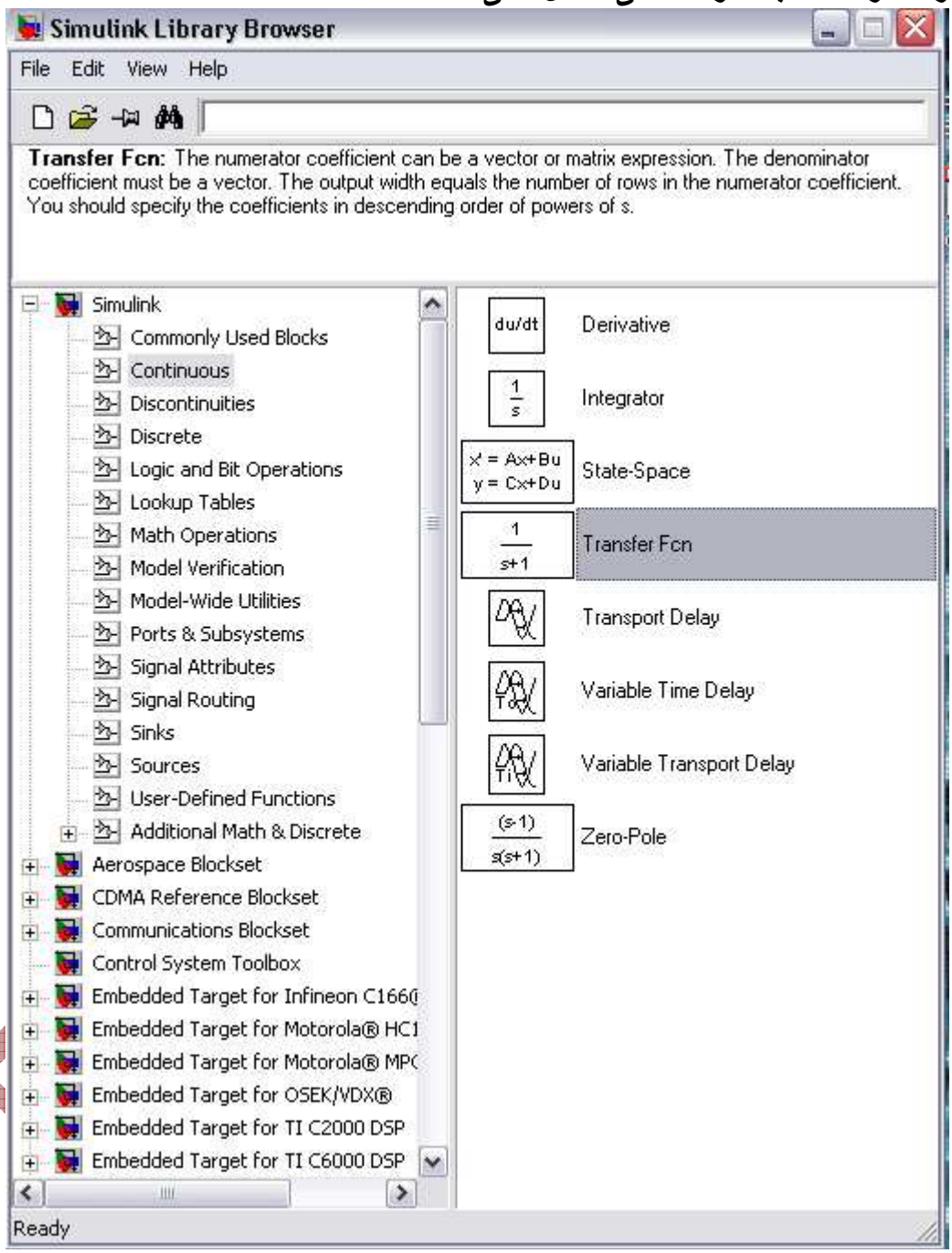
قم بالضغط على قائمة simulinkلتفتح لك مجموعة من الاقسام

اختار منها continuous

ومن اليمين اختار transfer function

وقم بالضغط على الماوس وسحب البلوك الى النموذج

والخطوات السابقة موضحة فى الشكل الاتى



رابعا ::لاضافة ال Mux block

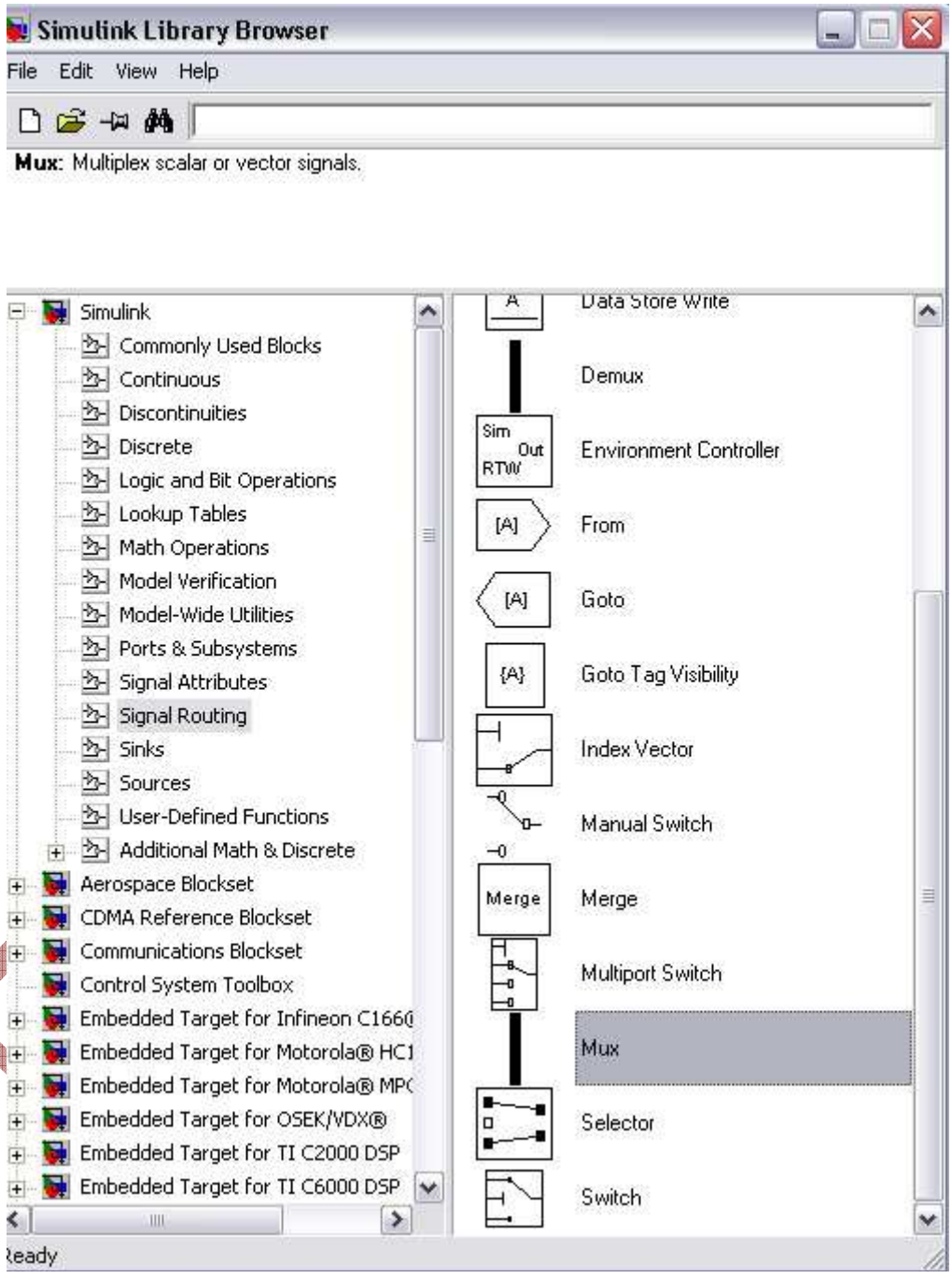
قم بالضغط على قائمة simulinkلتفتح لك مجموعة من الاقسام

اختار منها Signal Routing

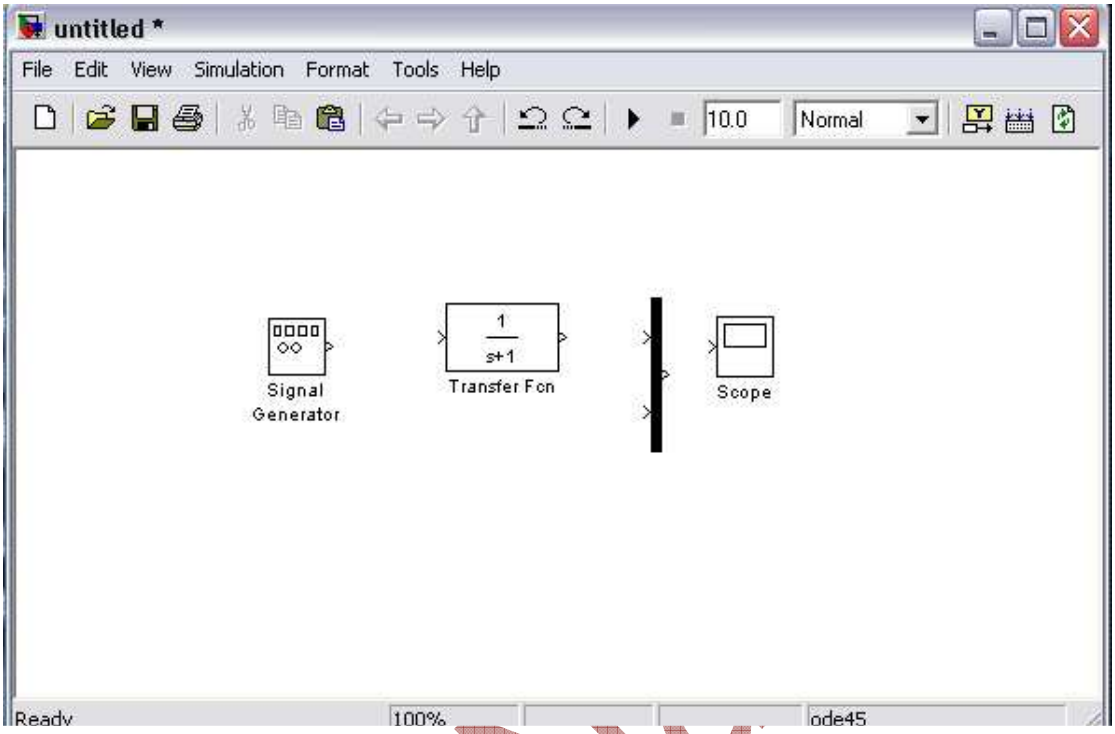
ومن اليمين اختارMux

وقم بالضغط على الماوس وسحب البلوك الى النموذج

والخطوات السابقة موضحة فى الشكل الاتى



والان سيكون شكل النموذج لدينا كما يلى ::



والان سبدأ فى عملية التوصيل بين البلوكات

::

وهناك طرقتين للتوصيل

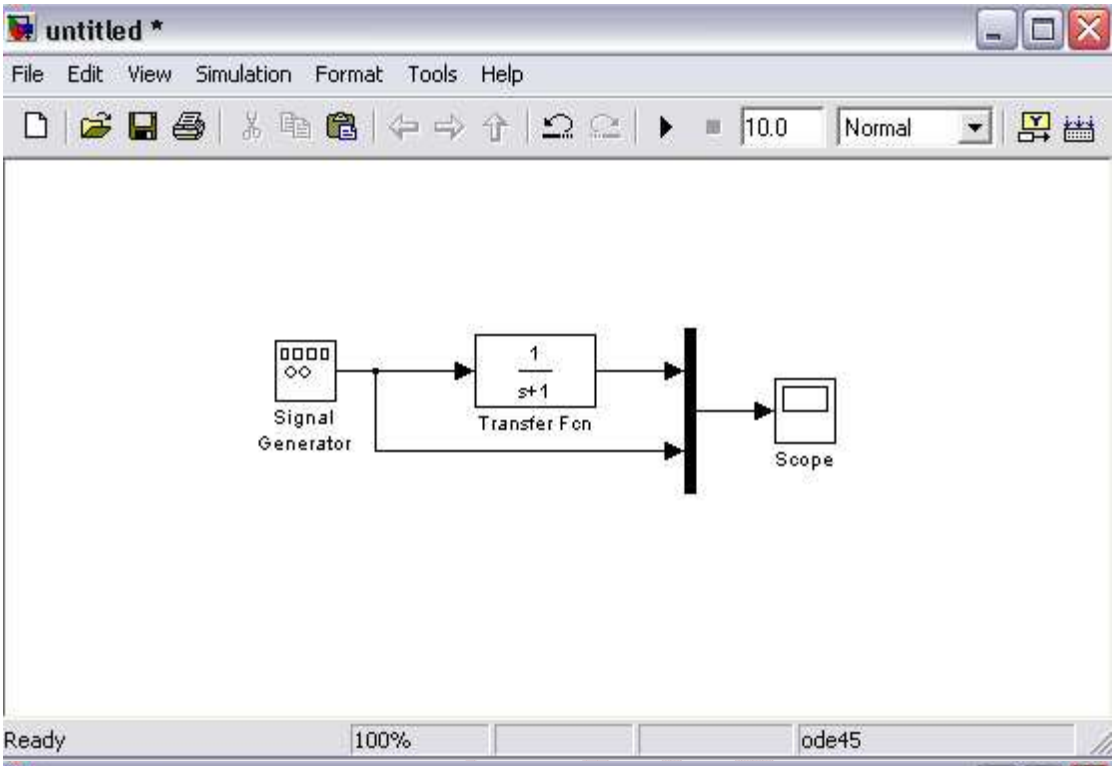
الاولى

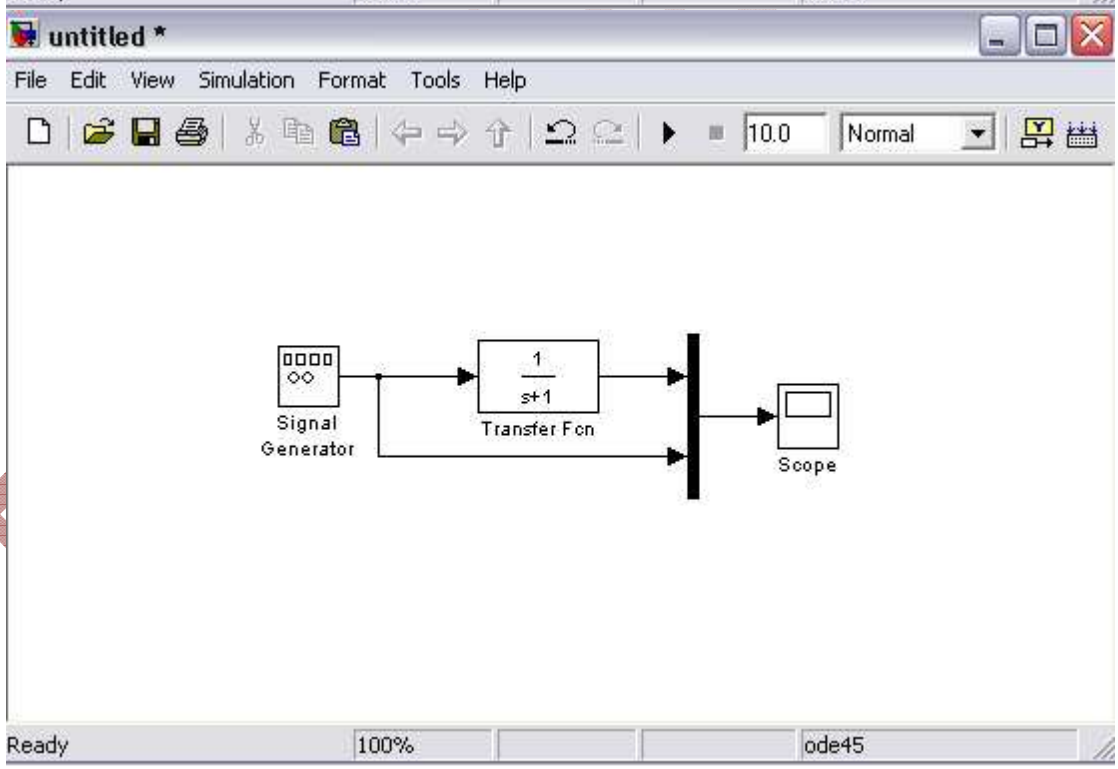
:

هى التوصيل السريع و للتوصيل المباشر و ليس للتوصيل الخطوط اى عمل العقد node وتكون عن طريق الضغط على البلوك المراد توصيله حتى يتم تحديده اى ظهور مربعات صغيرة من حوله وبعد ذلك قم بالضغط على مفتاح ctrl فى لوحة المفاتيح مع أستمرار الضغط قم بالذهاب بالماوس الى البلوك الاخر المراد توصيله و قم بالضغط عليه بالماوس ضغطة واحده و ستلاحظ ان يتم التوصيل مباشرة .

والطريقة الاخرى هى التوصيل اليدوى وتكون عن طريق تحريك الماوس على طرف البلوك عند النقطة المراد توصيلها حتى ياخد شكل الماوس علامة +وبعد ذلك قم بالضغط على الماوس و استمر فى الضغط وقم بالسحب حتى النقطة الاخرى المراد توصيلها حتى يأخد شكل الماوس علامة +ولكن مذدوجة ثم اترك الماوس ليتم التوصيل مباشرة كما يلى .

Eng.M.Nabil





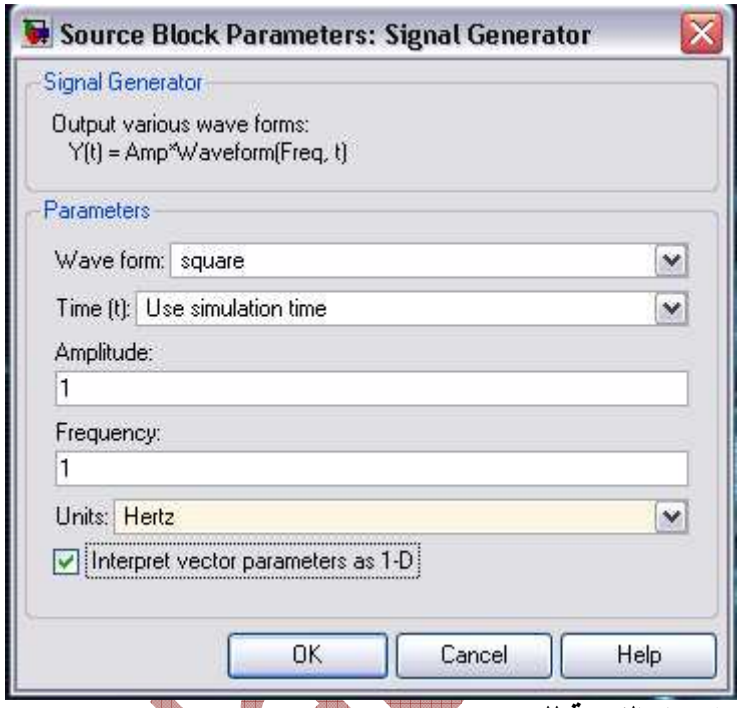
والان سنتقل الى مرحلة

Configuring the Model

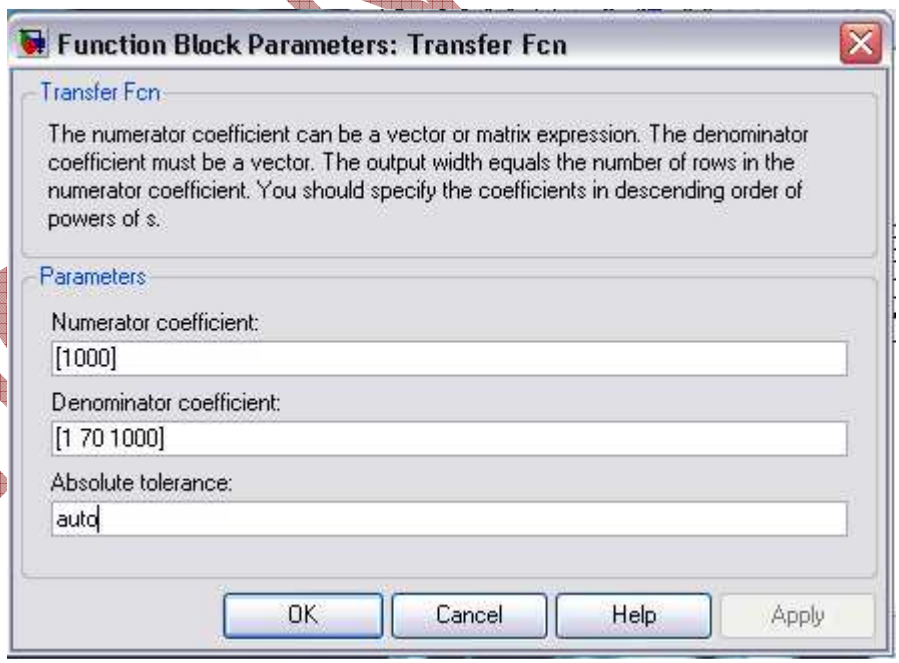
اى التعديل فى خصائص البلوكات

اولا

::قم بالضغط double click على بلوك signal generator وقم بوضع الخصائص كما موضح فى الصورة التالية ::



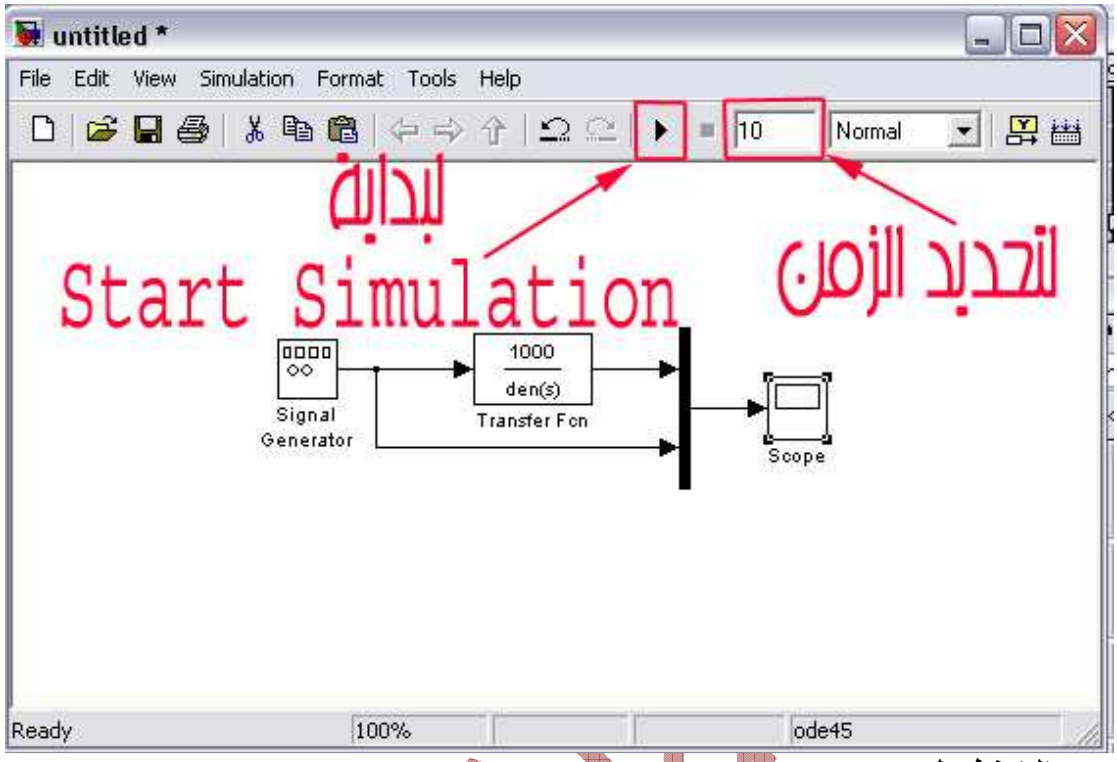
وايضا بالنسبة لل
transfer function



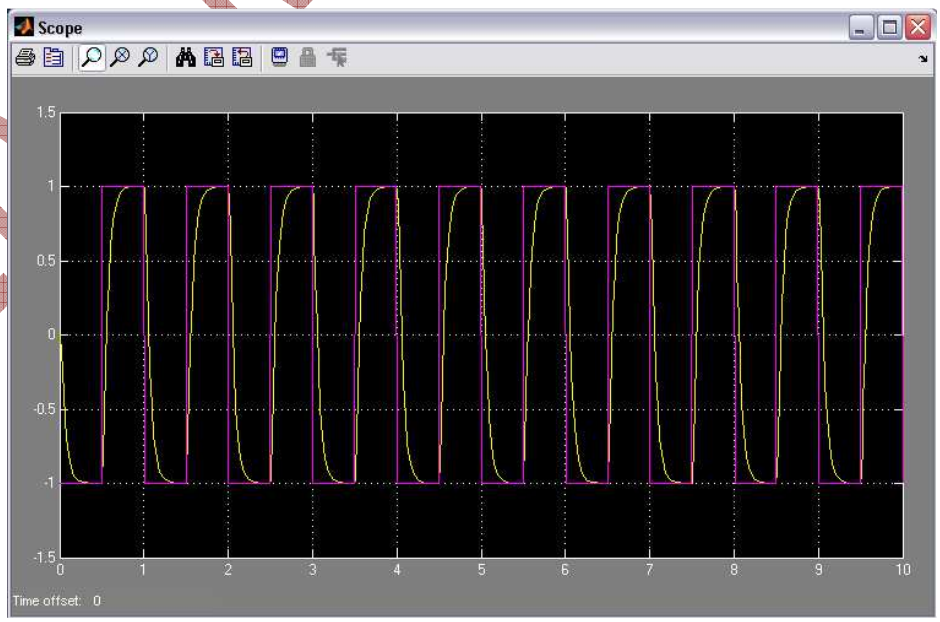
والان سنتقل الى مرحلة عمل ال Simulation

اولا قم بتحديد زمن ال simulationوليكن 10ثوانى

ثانيا
:قم بالضغط على start simulationكما موضح فى الشكل التالى :



وبعد الضغط على start simulation قم بالضغط double click على أل Scope لمشاهدة نتيجة عملية أل Simulation وهى عبارة عن استجابة النظام الموجود وهو نظام من الدرجة الثانية ل square wave لمدة 10ثوانى كما موجود فى الصورة الاتية



كيف يعمل السميولينك ؟؟

مقدمة

يقوم السميولنك بمحاكاة الأنظمة الديناميكية كما تعرفنا سابقا وتتم هذه العملية بمرحلتين هما:
المرحلة الأولى يقوم بها المستخدم بعمل النموذج والذى يحتوى على مجموعة البلوكات المطلوبة.
والمرحلة الثانية يقوم البرنامج بتنفيذ عملية المحاكاة فى الفترة الزمنية المطلوبة.

نمذجة المنظومات الديناميكية

من المعروف ان المنظومات الديناميكية تتكون من مجموعة من المعادلات الرياضية ويتم تمثيل هذه المعادلات فى السميولنك على هيئة بلوكات وهذه الفكرة مأخوذة من مبادى التحكم الالى والمعروف بى
Block Diagram
وتنقسم البلوكات فى السميولنك الى نوعان نوع افتراضى و نوع غير أفتراضى (nonvirtual block and virtual blocks) .
الانواع الغير افتراضية هى التى تمثل عناصر النظام الديناميكى اما الانواع الافتراضية وهى التى تستخدم فى تحويل الاشارات وغيرها دون ان تدخل فى تكوين النظام او معادلاته الرياضية .

ما معنى "time-based block diagram"؟؟؟

1-اى ان هناك علاقة زمنية بين الاشارات وبين المتغيرات (state variables)ويكون حل النموذج او block diagramهو حل لهذه العلاقات خلال الزمن المحدد time step والذى يمثل بزمن البداية الى زمن النهاية .
2-الاشارات تعبر عن كميات تتغير مع الزمن وتكون معرفة خلال الفترة الزمنية المحددة.
3-العلاقة بين الاشارات والمتغيرات تكون عبارة عن مجموعة من المعادلات اى ان كل بلوك يحتوى على مجموعة من المعادلات وهذه المعادلات توضح العلاقة بينه و بين الداخل له وبين الخارج منه.

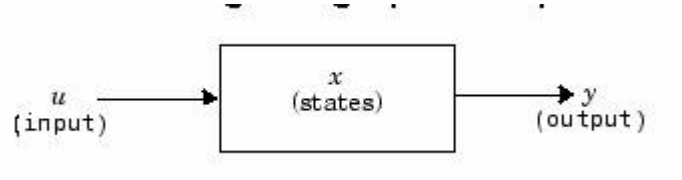
ويوجد نوعان من انواع البلوكات تبعا لنوعية بناءها

1-البلوكات الموجودة فى البرنامج وتسمى built-in blocks
2-البلوكات التى يقوم المستخدم بعملها وتسمى
custom blocks User-defined

States

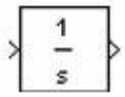
وهى التى تمثل قيم النظام الموجود وهى عبارة عن مجموعة من المتغيرات والتى تستخدم فى حساب الخرج الخاص بالبلوك عند الخطوة الزمنية المحدده وهناك نوعان من انواع ال states وهما Discrete : و continuousمتقطعة ومستمرة.
والمستمرة هى التى تتغير باستمرار اما المتقطعة هى التى تتغير عند فترات زمنية محددة intervals .
وتعتبر البلوكات States كما فى الشكل التالى

Eng.M.Nabil



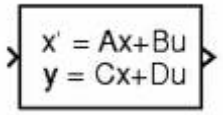
والبلوكات التى تعبر continuous states يجب ان تحتوى على احد البلوكات الاتية :

- Integrator



ووظيفة هذا البلوك هو تكامل الاشارة الداخلة

- State-Space



ووظيفة هذا البلوك هو عمل نظام خطى من النوع State-Space

محاكاة الانظمة الديناميكية

ماذا يحدث عند الضغط على Start Simulation ؟؟؟

اولا Model Compilation

اى يقوم السيميولنك بتحويل البرنامج او النموذج الى الصورة التى يمكنه حلها executable form
ويكون تسلسل خطوات هذه العملية كما يلى :

1. **حساب قيم التعبيرية لخصائص البلوكات وايجاد قيمتها block parameter expressions**
2. **تحديد خصائص الاشارات مثل نوعها و نوع البيانات و ابعادها**
3. **ويقوم بتحديد الخصائص الغير موجودة ويستخدم طريقة attribute propagation**
4. **يقوم بعملية تخفيض للبلوكات optimizations**

Eng.M.Nabil

5. يقوم وضع النموذج فى تسلسل هرمى
hierarchy

6. يقوم بتحديد رتبة كل بلوك وسوف نتعرض له لاحقا
block sorted order

7. يقوم بتحديد الفترات الزمنية لكل بلوك
Sample Time

ثانيا Link Phase :

وفى هذه المرحلة يقوم البرنامج بتحديد الاماكن اللازمة فى الذاكرة لتنفيذ النموذج

ثالثاSimulation Loop Phase :

وفى هذه المرحلة يقوم البرنامج بحساب قيم الحالات والخرج خلال الفترات الزمنية حتى نهاية زمن المحاكاه

وتحتوى هذه المرحلة على مرحلتين فرعيتين:

1- Loop Initialization phase

وهذا يحث مرة واحده فقط فى البداية

2- Loop Iteration phase

اما هذه المرحلة فيعاد تكرارها عند بداية كل فترة زمنية جديدة ويحدث بها

- حساب خرج البرنامج او النموذج
- حساب حالة البرنامج او النموذج

- البحث عن حالات غير مستمرة فى البلوكات المستمرة باستخدام
zero-crossing detection
(هذه الخطوة أختيارية) وسوف نتعرض لها لاحقا

- حساب زمن الفترة الزمنية التالية
ويتم تكرار هذه الخطوات طوال زمن المحاكاة .

طرق الحل فى السميولنك

حل النموذج المقصود به هو عملية حساب الحالات المتعاقبة
successive states
وطرق الحل هى عبارة عن مجموعة من البرامج الموجودة فى البرنامج وتسمى
solvers
ومن أهم الانواع

Eng.M.Nabil

1- Fixed-step solvers

وهى التى تقوم بحل النموذج فى فترات زمنية منتظمة من البداية حتى النهاية وحجم هذه الفترات يعرف بى step size وكما نعلم مع تقليل ال step size فان الدقة سوف تزيد.

2- Variable-step solvers

وهى التى تقوم بحل النموذج فى فترات زمنية متغيرة فتقوم بتصغير حجم الفترة الزمنية لزيادة الدقة عندما تتغير حالة النموذج بسرعة وتقوم بتكبير حجم الفترة الزمنية عندما يكون التغير فى الحالة بطىء.

وهناك ايضا من انواع Solvers تبعا للحالة مثل

1- Continuous solvers

وتكون عبارة عن عملية تكامل عددى فى الفترة الزمنية الحالية لحساب حالة النموذج من الحالة عند الفترة الزمنية السابقة ومن مشتقتها كما ذكر سابقا.

2- Discrete solvers

وتقوم بحساب حجم الفترة الزمنية التالية فقط

Zero-Crossing Detection

وهى الطريقة المستخدمة فى البحث عن الحالات الغير المستمرة فى كل فترة زمنية وعندما يجد البرنامج منطقة بها عدم استمرارية يقوم بتحديد زمنها بدقة ويقوم بأخذ فترات زمنية أضافية قبلها وبعدها .

لماذا نستخدم هذه الطريقة ؟؟

وذلك لان عند مناطق عدم الاستمرارية يحدث تغير هام جدا فى الخصائص الديناميكية للنظام وغالبا تتزامن مناطق عدم الاستمرارية مع الاحداث المهمة فى النظام .
ومن الممكن الاستغناء عن هذه الطريقة بتقليل حجم الفترة الزمنية الى قيم صغيرة جدا مما قد يودى الى زيادة زمن المحاكاة.

Algebraic Loops

بعض البلوكات فى السميولنك لديها مداخل تسمى

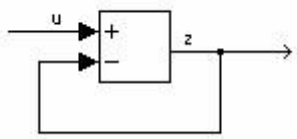
direct feedthrough

وهذا معناه ان خارج هذه البلوكات لا يمكن حسابه بدون معرفة قيم الاشارات الداخلة ومن أهم هذه البلوكات :

- The Math Function block
- The Gain block
- The Integrator block's initial condition ports
- The Product block
- The State-Space block when there is a nonzero D matrix
- The Sum block
- The Transfer Fcn block when the numerator and denominator are of the same order
- The Zero-Pole block when there are as many zeros as poles

Eng.M.Nabil

وال algebraic loop يحدث عندما يكون الداخل
direct feedthrough
وأيضا معرض للخارج كما فى الشكل التالى :



ومن الشكل السابق نرى ان z = u – z ويكون الحل z = u/2
ويمكننا أيضا استخدام بلوك
Algebraic Constraint
فى عمل algebraic loop

Modeling and Simulating Discrete Systems

تكمن مقدرة السميولنك على محاكاة الانظمة ذات الزمن المتقطع
والتى تسمى (sampled data) وايضا قدرة على محاكاة الانظمة التى يكون معدل تغيرها
غير ثابت - (multirate systems) والتى يكون فيها بلوكات ذات فترة زمنية معينة
وبلوكات أخرى ذات فترة زمنية مختلفة - ومحاكاة الانظمة التى تجمع بيانات متصلة و متقطعة
معا (hybrid systems) فى الخاصيتين التاليتين::

1- SampleTime block parameter

يوجد نوعان من نوع حجم الفترة الزمنية block parameter Sample Time وهما
explicit و implicit والبلوكات ذات الزمن المتصل تكون من النوع. implicit

2- Sample-time inheritance

يمكن لأغلب بلوكات السميولنك ان تاخد حجم الفترة الزمنية الخاص بها من البلوك المتصل
بمدخلها أما بالنسبة للبلوكات التى ليس لها مدخل يمكنها ان تتوارث الفترة الزمنية من
البلوكات المتصلة بمخارجها .

Determining Step Size for Discrete Systems

يقوم السميولنك باختيار حجم للفترة الزمنية
step size
متزامن مع الزمن الخاص بمعدل تقطيع الاشارة sample time hits
ويكون أختياره بناء على fundamental sample time
ويكون fundamental sample time هو أكبر عدد صحيح مقسوما عليه ال sample
timeمثلا:

لدينا sample times 0.25 و 0.5 فيكون

Eng.M.Nabil

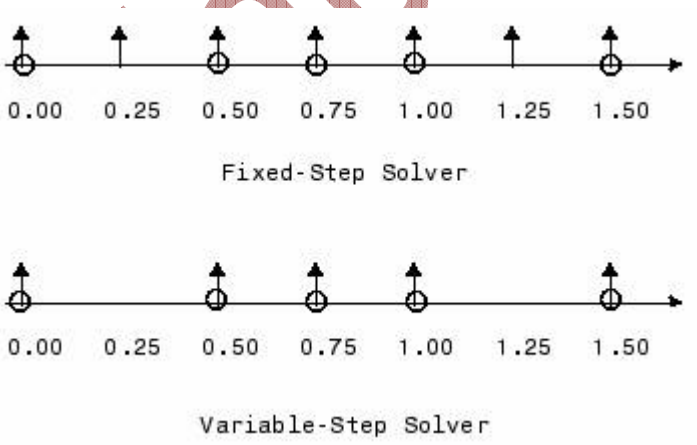
fundamental sample time 0.25

ويمكننا فى محاكاة الانظمة ذات الزمن المتقطع استخدام كلا النوعين من ال solver وهما
variable-step discrete

او fixed-step

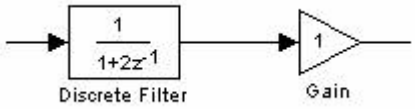
وفى حالة fixed-step يكون simulation step size يساوى fundamental sample time

وفى حالة variable-step solver يكون step sizeمساوى للمسافة بين نقط التقطيع sample time hits.
والفرق بينهم موضح فى الشكل التالى :



Sample Time Propagation

توليد زمن التقطيع أو حجم الفترة الزمنية
يقوم السميولينك هذه العملية فى بداية المحاكاة لتحديد زمن التقطيع للبلوكات التى تتوارث زمن تقطيعها وهى البلوكات التى ليس لها مداخل ومثلا فى النموذج الاتى ::



نرى بلوك gain ووظيفته هى ضرب الدخل فى ثابت والناتج يكون هو الخرج ولذلك الخارج يكون له نفس زمن تقطيع البلوك السابق له ويقوم السميولينك بالاتى ::
1-اذا كان الداخل له نفس زمن التقطيع فان السميولينك يقوم بتخصيصه
2 -اذا كان الداخل له زمن تقطيع مختلف ولكن عدد صحيح و اسرع من زمن البلوك نفسه فان السميولينك يقوم بتخصيص الزمن الاسرع

Constant Sample Time

زمن التقطيع الثابت::
والمقصود به هو زمن التقطيع الخاص بالبلوكات التى لا يتغير زمن تقطيعها اثناء عملية المحاكاة وشروط هذه البلوكات::

1-ان يكون جميع معاملات البلوك parameters غير قابلة للتعديل أثناء المحاكاة nontunable
2-ومن الممكن وضع زمن التقطيع لهذه البلوكات مالانهاية (inf) او تكون قابلة لتوارث زمن التقطيع من بلوكات أخرى بشرط ان تكون تلك البلوكات ذات زمن تقطيع ثابت أثناء عملية المحاكاة .
ويقوم السميولينك عمل بحث عن هذه البلوكات قبل بداية المحاكاة حتى يسهل من عملية الحسابات

أما اذا وجد السميولينك بلوكات لديها زمن تقطيع مالانهاية ولكن لا تعتبر ذات زمن ثابت وذلك نتيجة وجود معاملات من الممكن تعديلها Tunable Parameters
فانه يقوم بتنفيذ عملية Sample Time Propagation وقد سبق ذكرها

انتظروا المزيد.....
والحمد لله رب العالمين

م/محمد نبيل /معيد بكلية الهندسة جامعة كفرالشيخ
Eng_nabil_it@yahoo.com

Eng.M.Nabil